Anonymous Credentials Light

Foteini Baldimtsi, Anna Lysyanskaya
foteini,anna@cs.brown.edu

Computer Science Department, Brown University

Caroline Sandsbråten

What is the paper about?

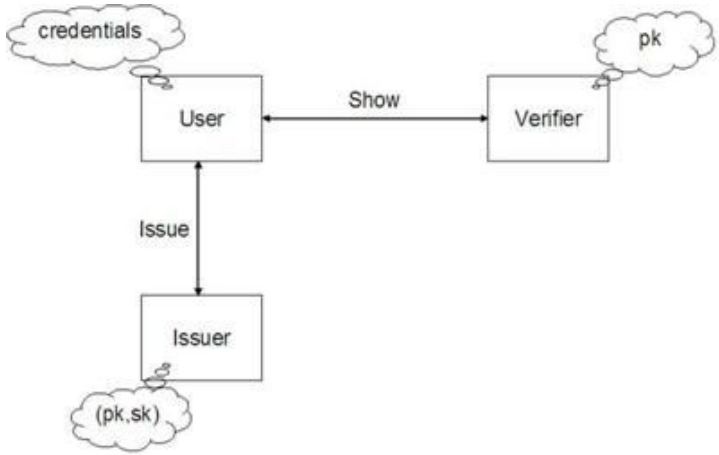
- Define and propose an efficient and provable secure construction of blind signatures with attributes
  - A building block for anonymous credential systems
- First provably secure construction of anonymous credentials in the elliptic group setting without bilinear pairings based on the DDH assumption
  - Previous work is based on the RSA group or on groups with pairings
- Only prior efficient construction that could work in such elliptic curve groups does not have a proof of security

Anonymous Credentials

- Allow users to prove possession of credentials without revealing information
- The proof is unlinkable to previous uses of the same credential
- Enable users to privately obtain credentials
- In essence: privacy preserving method that allows for verifying credentials without compromising a users anonymity

Anonymous Credentials: Why?

- Privacy preservation
  - Users don’t have to reveal their identity
  - Supports minimal disclosure
- Identity management
  - Allows for managing multiple identities
  - Makes possible selective disclosure of identity attributes
- Security and trust
  - Provides a secure method for verifying identity without disclosing personal data
  - Trust is built by verifying credentials
  - Trust is built by not having to disclose personal data
- Minimization of personal data
  - Aligns with privacy standards
  - Why should transactions require more personal information than necessary?
- User empowerment
  - Gives users control over their own information
- Real-world applications
  - Useful in applications where ID verification is necessary and tracking is not



Anonymous Credentials

- 1982: David Chaum introduces the concept of blind signatures for untraceable payments, laying the groundwork for anonymous credentials.
- 2001: Jan Camenisch and Anna Lysyanskaya develop the first practical anonymous credential system, which includes the ability to revoke anonymity.
- 2004: Camenisch and Lysyanskaya introduce signature schemes and anonymous credentials from bilinear maps, improving the efficiency and security of anonymous credentials.
- 2005: Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya propose compact e-cash, which is a direct application of single-use anonymous credentials.
- 2008: Jan Camenisch and Thomas Groß make advancements with efficient attributes for anonymous credentials, enhancing the practicality of the system.
- 2012: This paper: A new efficient and secure construction of blind signatures with attributes, optimized for lightweight devices like mobile phones, RFIDs, and smartcards.

Limitations of Previous Systems

- RSA group limitations:
  - Large key sizes to be secure
  - Exponentiation is expensive
  - NOT suited for low-power devices
- Bilinear pairings
  - Large security parameters required
  - Also computationally expensive
- Hardware accelerators can improve performance
  - But is not very practical :(
- All-in-all:
  - Previous solutions not practical when fast authentication or battery-powered devices are used

Blind Signatures

- 2 parties, signer and user
- Signer signs a message without knowing the message

> Thus, a better question is: how do we extend the notion of a blind signature so that it can efficiently accommodate anonymous credentials — if not full anonymous credentials a-la Camenisch and Lysyanskaya, then at least linkable lightweight anonymous credentials a-la Brands?

Blind Signatures with Attributes

- Extends blind signatures:
  - Incorporates attributes into the signing process
  - But what are attributes?
- Signer and user get as input:
  - Commitment C to the users attributes
- As output:
  - User obtains another unlinkable commitment C’ to the same attributes
  - A signature of this commitment
  - A message of the user’s choice
- User can prove that the C’ contains the correct attributes via a separate ZK proof

From Blind Signatures with Attributes to Anonymous Credentials

- Blind signatures with Attributes is the right building block for linkable anonymous credentials
  - User with some attributes forms a commitment C
  - Proves in ZK that she has committed to the correct set of attributes
  - Obtain credentials from the blind signatures with attributes protocol (sig, C’)
  - Can prove the credentials by revealing sig and ZK proof that C’ corresponds to the needed attributes

Anonymous Credentials Light (ACL)

- Three phases:
  - Registration
  - Preparation
  - Validation
- Efficiency:
  - Signer: 7 exponentiations
  - User: 13 exponentiations
  - Verification: 8 exponentiation
- Signature size is not explicitly mentioned
  - Paper suggests that it is relatively small compared to other schemes
  - I assume it would be about the same as the Abe blind signature

Pedersen Commitment

- Takes a set of messages and randomness R as input

1. *Setup*: On input the security parameter 1^k and the maximum number of messages n, pick a group G of prime order $q = \Theta(2^k)$ with generators $h, h_1, \ldots, h_n$.
2. *Commit*$(L_1, \ldots, L_n; R) = h^R \prod_{i=1}^{n} h_i^{L_i}$; note that $L_i \in \mathbb{Z}_q$.

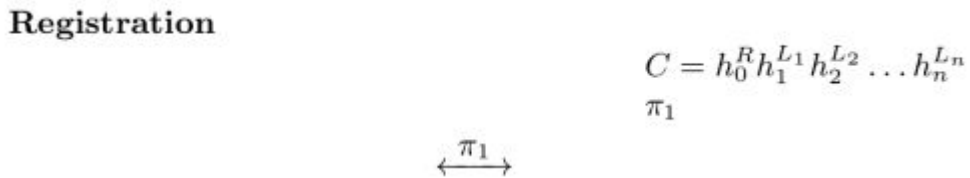
Combined/Blinded Pedersen Commitment

- Two commitments C_1 (to $L_1,...,L_n$), C_2 (to L_0) outputs the combination C
- Output Commit = (z^y, C^y)

Registration

Setup:

- Group G of order q, g is a generator, H a hash function to Z_q
- A trusted party that choose G, g and outputs params=(q,G,g,z,h,h_0,...,h_n)
- z, h, h_i are group elements of G
- n is max attributes
- Input to TP: G,q,g,h,y, outputs tag public key to signer G,q,g,h,y,z
- Signer public key: $y=g^x \bmod q$



Preparation

Preparation

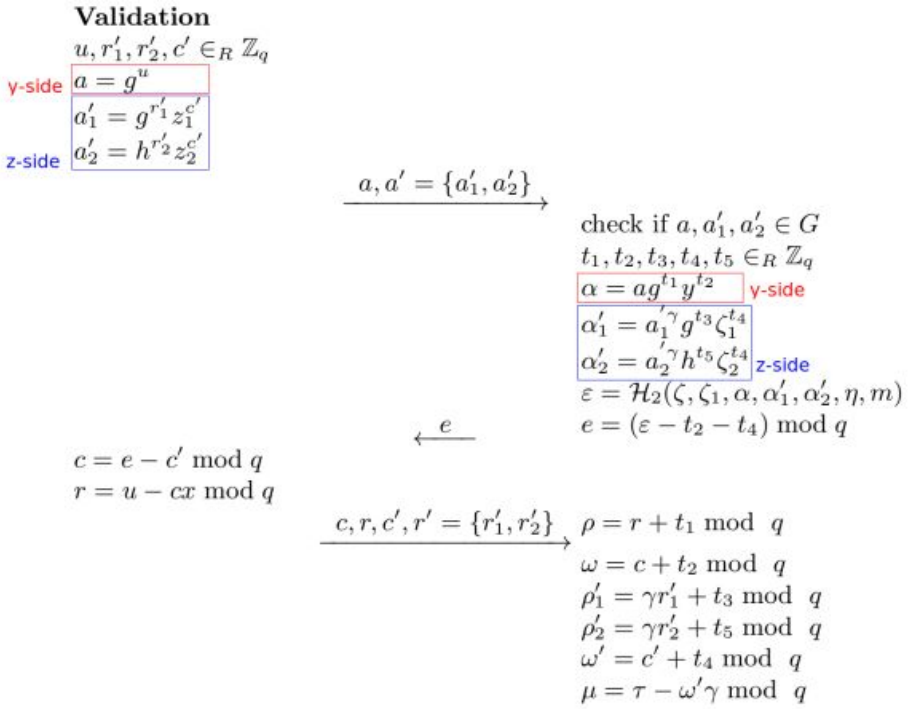
$rnd \in_R \mathbb{Z}_q$
$z_1 = Cg^{rnd}$
$z_2 = z/z_1$

$\xrightarrow{rnd}$

check if $rnd \neq 0$
$z_1 = Cg^{rnd}$
$\gamma \in_R \mathbb{Z}_q^*$
$\zeta = z^{\gamma}$
$\zeta_1 = z_1^{\gamma}$
$\zeta_2 = \zeta/\zeta_1$
$\tau \in_R \mathbb{Z}_q$
$\eta = z^{\tau}$

- y-side: proof of knowledge x of $y = g^x$
- z-side: proof of knowledge (w_1, w_2) of $z_1 = g^{w_1}$, $z_2 = h^{w_2}$ (where z_1, z_2 are the so called "one-time" tag keys that the signer creates).

Validation


Validation
$u, r_1', r_2', c' \in_R \mathbb{Z}_q$
y-side
$a = g^u$
z-side
$a_1' = g^{r_1'} z_1^{c'}$
$a_2' = h^{r_2'} z_2^{c'}$
$a, a' = \{a_1', a_2'\}$
check if $a, a_1', a_2' \in G$
$t_1, t_2, t_3, t_4, t_5 \in_R \mathbb{Z}_q$
$\alpha = a g^{t_1} y^{t_2}$
y-side
$\alpha_1' = a_1'^{\gamma} g^{t_3} \zeta_1^{t_4}$
$\alpha_2' = a_2'^{\gamma} h^{t_5} \zeta_2^{t_4}$
z-side
$\varepsilon = \mathcal{H}_2(\zeta, \zeta_1, \alpha, \alpha_1', \alpha_2', \eta, m)$
$e = (\varepsilon - t_2 - t_4) \bmod q$
e
$c = e - c' \bmod q$
$r = u - cx \bmod q$
$c, r, c', r' = \{r_1', r_2'\}$
$\rho = r + t_1 \bmod q$
$\omega = c + t_2 \bmod q$
$\rho_1' = \gamma r_1' + t_3 \bmod q$
$\rho_2' = \gamma r_2' + t_5 \bmod q$
$\omega' = c' + t_4 \bmod q$
$\mu = \tau - \omega' \gamma \bmod q$

ACL Construction

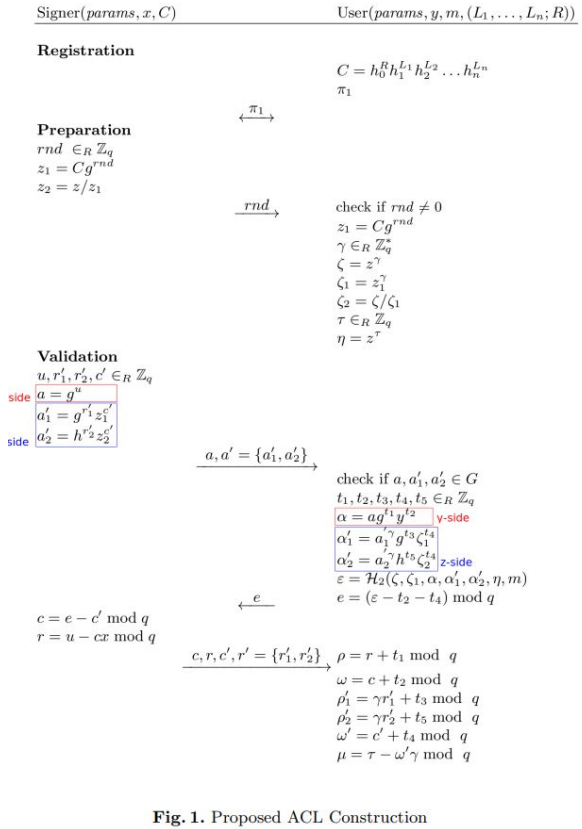


Fig. 1. Proposed ACL Construction

Security

- Blindness is satisfied under DDH assumption, in RO model
  - Given G of order q, with generator g, and g^a, g^b, g^c for $a,b,c \in Z_q$ it is infeasible to distinguish between (g,g^a,g^b,g^{ab}) and (g,g^a,g^b,g^c)
- Unforgeability is satisfied under Dlog
  - Let G of order q with generator g, then given g, h find exponent x s.t. $g^x = h \in G$

Blindness

Definition 6. *(Blindness for blind signatures with attributes) Let $\mathcal{A}$ be a malicious Signer and $b \in \{0,1\}$ be a randomly chosen bit which is kept secret from $\mathcal{A}$. $\mathcal{A}$ will try to guess the value b by performing the following steps:*

[noitemsep]

1. $(pk, sk) \leftarrow KeyGen(1^k)$
2. $\{m_0, m_1, \overrightarrow{L_0}, \overrightarrow{L_1}, R_0, R_1\} \leftarrow \mathcal{A}(1^k, pk, sk)$ *(i.e. $\mathcal{A}$ produces two messages $\{m_0, m_1\}$, polynomial in 1^k, and two attribute vectors $\overrightarrow{L_0}, \overrightarrow{L_1}$ with the corresponding randomness).*
3. $\mathcal{A}(1^k, pk, sk, m_0, m_1, \overrightarrow{L_0}, \overrightarrow{L_1}, R_0, R_1)$ *engages in two parallel (and arbitrarily interleaved as desired by $\mathcal{A}$) interactive protocols, the first with $U(pk, \{m_b, \overrightarrow{L_0}, R_0\})$ and the second with $U(pk, \{m_{1-b}, \overrightarrow{L_1}, R_1\})$.*
4. *$\mathcal{A}$ only gets signatures and the corresponding blinded commitments for the instances that didn't fail; for the ones that failed $\mathcal{A}$ gets nothing. If neither of the User instances failed, then $\mathcal{A}$ gets two signatures and the corresponding blinded commitments: $\sigma(m_0, \tilde{C}_b), \tilde{C}_b$ and $\sigma(m_1, \tilde{C}_{1-b}), \tilde{C}_{1-b}$.*
5. *$\mathcal{A}$ outputs a bit b'.*

Then the probability, taken over the choice of b, over coin-flips of the key-generation algorithm, the coin-flips of $\mathcal{A}$, and (private) coin-flips of both users (from step 3), that $b' = b$ is at most $\frac{1}{2} + \nu(k)$, where $\nu(k)$ is a negligible function.

We give the definition of one-more unforgeability for the sequential composition case.

Blindness

- When interacting with a challenger (Ch), the adversary (A) cannot guess which signature is which
- They prove this with the following 3 games:
  - Real: Ch outputs 2 correct signatures
  - Hybrid: Ch outputs 1 fake, 1 correct signature
  - Fake: Ch outputs 2 fake signatures
- Need to prove that Real=Hybrid=Fake

Unforgeability

Definition 7. *(Sequential one-more unforgeability for blind signatures with attributes) (KeyGen, BlindSign, Verify) is a one-more unforgeable blind signature scheme with respect to Commit if* $\forall$ *ppt* $\mathcal{A}$*, the probability that* $\mathcal{A}$ *wins in the following game is negligible:*

[noitemsep]

1. $(pk, sk) \leftarrow KeyGen(1^k)$
2. $\mathcal{A}(pk, params)$ *engages in polynomially many (in k) adaptive, sequential interactive protocols with polynomially many copies of the signer, where* $\mathcal{A}$ *decides in an adaptive fashion when to stop. For every execution,* $\mathcal{A}$ *forms a commitment* C_i *and picks a message* m_i *and invokes* $S(pk, sk, C_i)$*. Let* ℓ *be the number of executions, where the Signer output "completed" in the end of the protocol.*
3. $\mathcal{A}$ *outputs a collection* $\{(\tilde{C}_1, m_1, \sigma_1), \ldots, (\tilde{C}_j, m_j, \sigma_j)\}$ *where* $(\tilde{C}_i, m_i, \sigma_i)$ *for* $1 \leq i \leq j$ *are all accepted by* $Verify(pk, \tilde{C}_i, m_i, \sigma_i)$*, and all* $(\tilde{C}_i, m_i)$*'s are distinct.*

We say that $\mathcal{A}$ *wins the game if either:*

1. $j > \ell$ *(i.e.* $\mathcal{A}$ *outputs more* $(\tilde{C}, m, \sigma)$ *tuples than he received).*
2. $\mathcal{A}$ *opens the sets of commitments* $\{C_i\}$ *and* $\{\tilde{C}_i\}$ *and the resulting multisets do not match.*

Unforgeability

Theorem 4 (One-More Unforgeability). *The signature issuing protocol is $(\ell, \ell+1)-$unforgeable for polynomially bounded ℓ if the discrete logarithm problem is intractable and $\mathcal{H}$ is a random oracle.*

- y-side: proof of knowledge x of $y = g^x$
- z-side: proof of knowledge (w_1, w_2) of $z_1 = g^{w_1}$, $z_2 = h^{w_2}$ (where z_1, z_2 are the so called "one-time" tag keys that the signer creates).

- Proof outline:
  - Prove indistinguishability from z-side proof of knowledge
  - Prove restrictive blinding
  - Prove that is infeasible to create a valid signature without engaging in the issuing protocol with the legitimate signer
  - 2 + 3 → User engages in the signature issuing protocol x times and outputs x + 1 signatures, then, there exist at least two valid signatures linked to a particular run of the issuing protocol
  - Prove that a user that can produce two signatures from 1 run can be reduced to the dlog problem

Implementation

Implementation. In an independent to this work [29] our ACL scheme has been implemented using an NFC smartphone: the BlackBerry Bold 9900. The implementation has been tailored for payments in transportation systems where our ACL construction is being used as an e-cash scheme (which as we mentioned above is a direct application of "single-use" credentials). The implementation has been based on eliptic curves and the results are very promising. Briefly, the signing/issuing takes a total of 300 milliseconds including terminal, communication and smartphone execution time, while spending (i.e. verification of the signature) takes about 380 milliseconds when 2 attributes are revealed which would be less if not attributes where revealed.