Cryptography fact sheet about Idemix's basic proof techniques by G. Alpár (February 17, 2014)

Schnorr's proof of knowledge: Given a group G of prime order q, in which the discrete logarithm (DL) problem is hard, and a generator g (so $\langle g \rangle = G$), a prover proves to a verifier that she knows a secret value $x \in \mathbb{Z}_q^*$ corresponding to a public value $h = g^x \in G$.

Prover Secret: x	$q, g, h = g^x$	**Verifier**
$w \in_R \mathbb{Z}_q^*$		
$a := g^w$ in G	$\xrightarrow{a}$	
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_q$
$r := c \cdot x + w \pmod q$	$\xrightarrow{r}$	$a \stackrel{?}{=} g^r \cdot h^{-c}$ in G

Schnorr's proof of knowledge for composite group order: Given the group $\mathbb{Z}_n^*$ of order $\varphi(n)$ where n is the product of two large prime numbers p, q. A prover proves to a verifier that she knows a secret value $x \in \mathbb{Z}_n$ corresponding to a public value $h = g^x \in \mathbb{Z}_n$. (The knowledge of $\varphi(n)$, p, or q is equivalent and we assume that at least the prover does not know these values.)

Prover Secret: x	$n, g, h = g^x$	**Verifier**
$w \in_R \{0,1\}^{\ell_n + \ell_c + \ell_\varnothing}$		
$a := g^w \pmod n$	$\xrightarrow{a}$	
	$\xleftarrow{c}$	$c \in_R \{0,1\}^{\ell_c}$
$r := c \cdot x + w$	$\xrightarrow{r}$	$a \stackrel{?}{\equiv} g^r \cdot h^{-c} \pmod n$

Bit-lengths in case of Idemix of the modulus n, the challenge c and the general security parameter are $\ell_n = 2048$, $\ell_c = 256$ and $\ell_\varnothing = 80$, respectively.

RSA signature: (relies on the RSA problem)

The signer's public key is (n, e) and his secret key is (p, q) where $n = pq$ and $e \in \mathbb{Z}_n^*$. Computations in the exponent are carried out modulo $\varphi(n) = (p-1)(q-1)$ which is only known by the signer. Then the RSA signature is

$$\sigma \text{ on } m, \qquad \text{where } \sigma \equiv m^{1/e} \pmod n$$

Verification:

$$m \stackrel{?}{\equiv} \sigma^e \pmod n.$$

RSA problem: Informally, find e^{th} root (i.e., given (b, e, n), compute a where $a^e \equiv b \pmod n$).

Didactic steps (without details):

(1) The composite Schnorr proof above, denoted abstractly as $PK\{(\alpha) : y = g^\alpha \pmod n\}$, is the simplest proof of knowledge in Idemix.

(2) The next step is to create a proof that a secret value (α) resides in a given interval $[a, b]$. (Here we don't give the details of this proof.)

(3) Then proofs can be combined, e.g., $PK\{(\alpha, \beta) : y = g^\alpha \ \wedge \ z = h_1^\alpha h_2^\beta \pmod n \ \wedge \ \alpha \in [a, b]\}$.

(4) Finally, by using the Fiat–Shamir heuristic, a proof can be executed in a non-interactive way as the challenge is the hash of the commitment a. (Note that the randomness of the challenge relies on the hash function.) If the input to the hash includes also a message m, one can also create a signature $\sigma(m) = (a, \mathcal{H}(a \| m), r)$. This is often used for signing a nonce ν (as the message) to prove freshness, e.g., $SPK\{(\alpha, \beta) : y = g^\alpha \wedge z = h_1^\alpha h_2^\beta \pmod n \wedge \alpha \in [a, b]\}(\nu)$.

Proof of knowledge (interactive or non-interactive) is the most fundamental technique to perform selective disclosure with attribute-based credentials.

Camenisch–Lysyanskaya (CL) signature on a single message and on multiple messages:

Let $p', q', p = 2p' + 1, q = 2q' + 1$ be large prime numbers. The system parameters are $n = pq$, $Z, S, R \in QR_n$ (multiplicative subgroup of quadratic residues: $QR_n \leq \mathbb{Z}_N^*$), the signer's secret key is (p, q), and the message is m. Then the CL signature is

$$(A, e, v) \text{ on } m, \qquad \text{where } A \equiv \left(\frac{Z}{S^v R^m} \right)^{1/e} \pmod n$$

where e, v are random, e is prime, and $\frac{1}{e} \cdot e \equiv 1 \pmod{\varphi(n)}$. Verification:

$$Z \stackrel{?}{\equiv} A^e S^v R^m \pmod{n}$$

In case of multiple attributes (message blocks), the system parameters are $Z, S, R_0, \ldots, R_l \in QR_n$, the signer's secret remains (p, q). Then the CL signature is

$$(A, e, v) \text{ on } (m_0, m_1, \ldots, m_l) \qquad A \equiv \left(\frac{Z}{S^v \prod_{i=0}^{l} R_i^{m_i}} \right)^{1/e} \pmod{n}$$

Verification:

$$Z \stackrel{?}{\equiv} A^e S^v \textstyle\prod_{i=0}^{l} R_i^{m_i} \pmod{n}$$

The CL signature relies on the strong RSA assumption that states that the following problem is hard (essentially, the RSA and the DL are both hard):

Strong RSA problem: Informally, find *any* root (i.e., given (a, n), find (b, c) where $c^b \equiv a \pmod{n}$).

Randomized Camenisch–Lysyanskaya signature:

One can create randomized versions of a CL signature: (A, e, v) becomes $(A', e, \hat{v})$ where $A' := A \cdot S^{-r} \pmod{n}$, $\hat{v} := v + er$ for any $r \in_R \{0,1\}^{\ell_n + \ell_\varnothing}$. Verification is done in the same way as without the randomization since

$$A'^e S^{\hat{v}} \textstyle\prod_{i=0}^{l} R_i^{m_i} \equiv A^e S^{-er} S^v S^{er} \prod_{i=0}^{l} R_i^{m_i} \equiv A^e S^v \prod_{i=0}^{l} R_i^{m_i} \equiv Z \pmod{n}.$$

Remark: This method does not yet provide unlinkability for a credential owner when showing randomized versions of a signature, since e remains the same; thus, the prover doesn't disclose e explicitly but proves to know it (see at Selective disclosure below).

Idemix credential issuing: signature on secret key m_0 and attributes $(m_1, \ldots, m_l)$

When CL-signatures are used as credentials, the roles of a signer (issuer) and the prover (credential owner or user) are separate. Only the issuer, knowing the prime factors of n, can produce the signature; however, by keeping the messages secret (the secret key m_0 and the attributes $m_1, \ldots, m_l$), a user actually knows a representation $(m_0, \ldots, m_l)$ of $\frac{Z}{A^e S^v}$ with respect to $(R_0, \ldots, R_l)$. He can then prove the knowledge of the attributes $m_1, \ldots, m_l$ or show any subset of them.

The issuing protocol is a blind signature in the sense that the user's secret key m_0 and v from the final signature (A, e, v) are only known to the user and not by the issuer. (The protocol is described without interval proofs, proof verifications, $\pm$, and freshness.)

Issuer Secret: p, q	Sys. pars. $(m_1, \ldots, m_l)$	**User** Secret: m_0
		Random v' $U := S^{v'} R_0^{m_0} \pmod{n}$
	$\xleftarrow{U, PK}$	$PK\{(\nu', \mu_0) : U \equiv S^{\nu'} R_0^{\mu_0} \pmod{n}\}$
Random v'' and prime e $A := \left(\frac{Z}{U S^{v''} \prod_1^l R_i^{m_i}} \right)^{1/e} \pmod{n}$		
$PK\{(\delta) : A \equiv \left(\frac{Z}{U S^{v''} \prod_1^l R_i^{m_i}} \right)^{\delta} \pmod{n}\}$	$\xrightarrow{(A, e, v''), PK}$	$v := v' + v''$ in signature (A, e, v)
		$Z \stackrel{?}{\equiv} A^e S^v \prod_0^l R_i^{m_i} \pmod{n}$

Idemix selective disclosure protocol

In the following example, the user discloses all but the first two attributes. That is, $m_3, \ldots, m_l$ are common input, while m_1, m_2 remain secret and private input for the user. Furthermore, the user does not reveal the signature (A, e, v) to the verifier, but first she randomizes it to $(A', e, \hat{v})$ and sends only A' to the verifier. Finally, she proves the knowledge of a corresponding valid signature, the secret key, and the hidden attributes. (The protocol is described without interval proofs, proof verification, $\pm$, and freshness.)

User Secret: $m_0, m_1, m_2, (A, e, v)$	Sys. pars. $(m_3, \ldots, m_l)$	**Verifier**
Random r : $(A' := A \cdot S^{-r}, e, \hat{v} := v + er)$		
$PK\{(\varepsilon, \hat{\nu}, \mu_0, \mu_1, \mu_2) : Z \prod_3^l R_i^{-m_i} \equiv A'^{\varepsilon} S^{\hat{\nu}} R_0^{\mu_0} \prod_{i=1}^{i=2} R_i^{\mu_i} \pmod{n}\}$	$\xrightarrow{A', PK}$	Verif.