A Description of Protocols for Private Credentials

Ariel Glenn, Ian Goldberg, Frédéric Légaré, Anton Stiglic
Zero-Knowledge Systems Inc.
{ariel, ian, frederic, anton}@zeroknowledge.com

1 Introduction

This document provides a short description of practical protocols for private credential systems.[1] We explain the basic concepts and mechanisms behind issuing and showing of private credentials and e-cash. The goal is to describe concisely how practical private credential systems can be achieved and not to provide intuition or motivation for the technology; for information on these subjects, see [1, 2, 3]. We give the details of one specific type of practical protocols for private credentials; other choices of functionalities and optimizations are possible. The reader is assumed to have general knowledge of basic concepts of cryptography such as the Discrete Logarithm problem, basic group theory and hash functions. For security proofs and more elaborate descriptions of the techniques used we refer the reader to [2].

2 Basic Tools

2.1 Notation

- $a \in A$ denotes that the value a is in the set A.
- $a \in_R A$ denotes the choice of a uniform and independent random value a from the set A.
- $\mathbb{Z}_p$ denotes the set $\{0, 1, \ldots, p-1\}$, $\mathbb{Z}_p^*$ denotes the set $\{1, \ldots, p-1\}$ for a prime p.
- $a \leftarrow b$ denotes the assigning to a of the value of b.
- $a := b$ denotes the definition of a as the value of b.

Let p be a large prime number, for which $p-1$ has a large prime factor q. (In practice, p will be around 1024 bits long, and q will be around 160 bits long.) For the rest of the paper, we assume operations in a subgroup of $\mathbb{Z}_p^*$ of order q, which we denote by $G_{q,p}$. All calculations involving elements of this subgroup are assumed to be mod p, and all calculations involving the field of exponents are assumed to be mod q. (Note that this is similar to the setup of DSA.)

[1] Note that this technology is patented.

2.2 DLREP function, DL-representation

Recall that the Discrete Log (DL) function in $G_{q,p}$ with respect to a generator g of $G_{q,p}$ is defined as a function of the form

$$f(x) := g^x \bmod p$$

where x is an element of $\mathbb{Z}_q$.

We will utilize a generalization of the DL function, called the **DLREP** function: A DLREP function with respect to elements $g_1, \ldots, g_l$, of $G_{q,p}$ is a function of the form

$$f(x_1, \ldots, x_l) := g_1^{x_1} \cdot \ldots \cdot g_l^{x_l} \bmod p$$

where $(x_1, \ldots, x_l)$ is a tuple whose elements are in $\mathbb{Z}_q$. This tuple is called a **DL-representation** of the product $h := g_1^{x_1} \cdot \ldots \cdot g_l^{x_l}$ with respect to the generators $(g_1, \ldots, g_l)$. For example, a trivial representation for $h = 1$ is $(0, \ldots, 0)$. When the context is clear, we simply call $(x_1, \ldots, x_l)$ a DL-representation of h.

2.3 Proving knowledge of a DL-representation

The primary tool we use for the protocols is the ability for a prover Peggy to demonstrate to a verifier Victor that she indeed knows some DL-representation of a number h, with respect to agreed-upon generators $(g_1, \ldots, g_l)$. Peggy of course does not want to reveal the DL-representation to Victor.

Suppose Peggy knows a DL-representation $(x_1, \ldots, x_l)$ of h, so $h = g_1^{x_1} \cdot \ldots \cdot g_l^{x_l}$. In order to prove this fact to Victor, she proceeds as follows:

- Peggy chooses l random elements, $w_1, \ldots, w_l$ in $\mathbb{Z}_q^*$ (she chooses a random DL-representation) and computes $a = \text{SHA1}(g_1^{w_1} \cdot \ldots \cdot g_l^{w_l})$.
- She then computes $c = \text{SHA1}(a, M) \bmod q$.
- Finally, she computes $r_i = c \cdot x_i + w_i$ for $i = 1, \ldots, l$.
- Peggy transmits $(a, r_1, \ldots, r_l)$ to Victor.

In the above protocol, M is an arbitrary message agreed upon by Peggy and Victor which should contain a nonce to ensure the freshness of c. It turns out that this proof of knowledge doubles as a digital signature algorithm: when Peggy proves to Victor that she knows a DL-representation of h, she also, at the same time and for free, produces a digital signature on M. (Viewed as a digital signature scheme, h is Peggy's public key, and $(x_1, \ldots, x_l)$ is her private key.)

In order for Victor to verify Peggy's proof (and signature on M), he first calculates c in the same way as Peggy, and then verifies that

$$a = \text{SHA1}(g_1^{r_1} \cdot \ldots \cdot g_l^{r_l} \cdot h^{-c})$$

Some notes on this protocol:

Security: This protocol is secure as long as at least one of the DLREP exponents is uniformly randomly chosen from $\mathbb{Z}_q^*$ and kept secret. That is, even if $x_1, \ldots, x_{l-1}$ are publicly revealed, the system *remains secure* as long as $x_l \in_R \mathbb{Z}_q^*$ and is known only to Peggy.

Lending protection: Peggy needs to *know* all of the x_i in order to complete this protocol, even if she doesn't *show* them all. Making one of the x_i be highly personal (or security-critical) information about Peggy (a password to a bank account, for example), will discourage Peggy from disclosing DL-representations she knows to others.

Reuse of the w_i: If Peggy ever uses the same value of w_i twice, for two different values of c, the resulting two (c, r_i) pairs can be used to recover x_i. We will use this fact to our advantage in section 6.

3 Private Credentials

In a private credential system, we have a credential authority (denoted by **CA**) who can certify certain attributes. Peggy executes an *issuing* protocol with the CA in order to obtain a credential that enables her to prove to Victor (through a *showing* protocol) that she has certain attributes, which are certified by the CA. The privacy requirements are twofold:

- Peggy should be able to reveal none, some, or all of her attributes during the showing protocol, and give no information about the attributes she did not reveal.
- The showing protocol should be *unlinkable* to the issuing protocol.

The second requirement means, for example, that if 50 people have been issued credentials with the "age" attribute having the value "under 13", then when such a credential is shown (say at a children's web-site), no collusion of participants (which does not involve the person doing the showing or all the other 49 people, of course) can figure out which of those 50 people it is.

We will now outline the main idea behind the scheme. In all of the computations, we will be working in a known group $G_{q,p}$, with a known generator g.

CA Setup: The CA secretly picks l random distinct elements of $\mathbb{Z}_q^*$: $y_0, y_1, \ldots, y_{l-1}$, and calculates (and publishes) the l values $h_0 = g^{y_0}, g_1 = g^{y_1}, \ldots, g_{l-1} = g^{y_{l-1}}$.

- The published values will be in $G_{q,p} \subset \mathbb{Z}_p^*$.
- The first published value is called h_0 instead of g_0 as it will play a different role from the rest of the g_i.

The published values form a public DLREP function:

$$f(\alpha, x_1, x_2, \ldots, x_{l-1}) := (g_1^{x_1} \cdot \ldots \cdot g_{l-1}^{x_{l-1}} \cdot h_0)^{\alpha} \bmod p$$

Note that although this function is slightly different from the original DLREP function described earlier, it is trivial to see that it is as secure.

Peggy's secret part: The secret part of Peggy's credential consists of a random value $\alpha \in_R \mathbb{Z}_q^*$, and (up to) $l-1$ attributes $x_1, x_2, \ldots, x_{l-1} \in \mathbb{Z}_q$. (Unused attributes can be set to 0.)

- The attributes might not *stay* secret, because Peggy may wish to show some or all of them as part of some showing protocol. However, the random value α must remain secret at all times.
- Note that the attributes are encoded as numbers in $\mathbb{Z}_q$. This encoding should be *unambiguous*; e.g. one should not be able to pass off an attribute with the value 20021231, which may encode an expiry date, for example, as an attribute representing income. There are two obvious ways to do this:

  Positional attributes: Each position is assigned an unambiguous meaning by the CA. For example, x_1 always represents expiry date, x_2 always represents birth year, etc. In this scenario, the values of the x_i can simply be small integers, or the numeric values of short strings.

  Labelled attributes: The positions of the attributes are irrelevant, but the value of any x_i is the numeric value of a string of the form "`LABEL:value`". Proving that x_2 = "`EXPIRY:20021231`" would have the same semantics as proving that x_5 = "`EXPIRY:20021231`".

  Positional attributes would seem to be a good choice when the set of attributes certifiable by a given CA is small and fixed; conversely, labelled attributes allows more flexibility.

Peggy's public part: The public part of Peggy's credential consists of the value of the DLREP function $h' = f(\alpha, x_1, \ldots, x_{l-1})$, along with a **restrictive blind signature** (u', v') on h' from the CA. Note that primed variables ($'$) denote blinded values.

Restrictive blind signatures With restrictive blind signatures, the signer gets to see certain parts of the structure of the message to be signed. This structure cannot be modified without invalidating the signature. In this application, the CA will get to see

the attributes $x_1, \ldots, x_{l-1}$, and will issue a credential only if those attributes are valid for Peggy.

As with public key certificates, it is necessary for Peggy to not only show a signature on h', but also to demonstrate knowledge of the secret part of the credential. In the particular protocol we present, it is possible for anyone to come up with a valid signature (u', v') on a random value h', but only the CA can create valid credentials for which Peggy knows the secret part.

4 Issuing Protocol

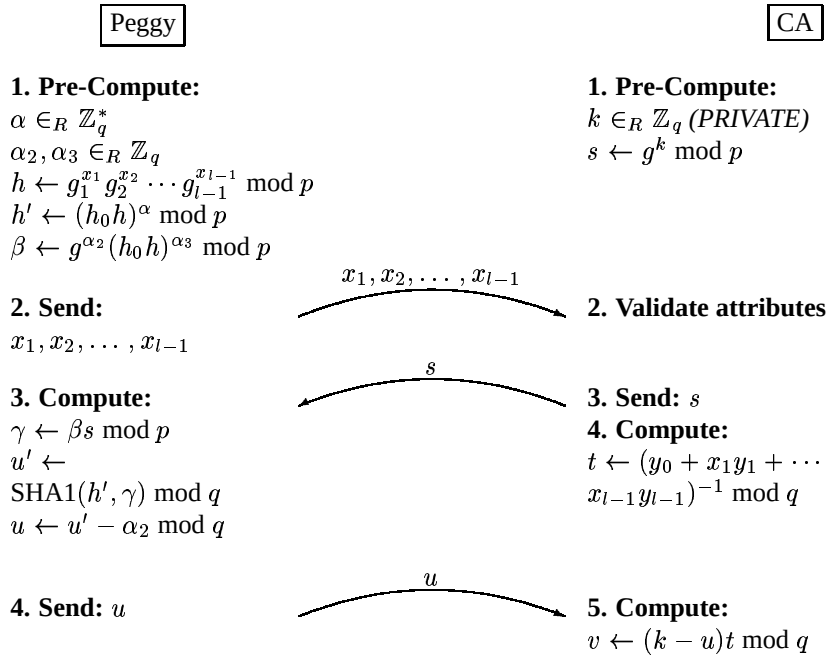
In this section, we describe the details of the protocol executed by Peggy and the CA, in order for the CA to issue a credential to Peggy. Recall that the CA has specified a group $G_{q,p}$ and a DLREP function

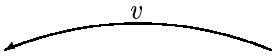
$$f(\alpha, x_1, x_2, \ldots, x_{l-1}) := (g_1^{x_1} \cdot \ldots \cdot g_{l-1}^{x_{l-1}} \cdot h_0)^{\alpha} \bmod p$$

Peggy has $l-1$ attributes $x_1, \ldots, x_{l-1}$ that she wants the CA to certify. At the end of the protocol, Peggy will have a random number α and a restrictive blind signature (u', v') on $h' = f(\alpha, x_1, x_2, \ldots, x_{l-1})$.

The best way to think about this protocol is that Peggy and the CA perform a *joint computation* of (u', v'), in such a way that Peggy never learns the CA's private key $(y_0, \ldots, y_{l-1})$, and the CA never learns α, h', u', or v'; they each compute the components of it that they can, and Peggy combines the results at the end.

The protocol proceeds as follows (usually through a secure authenticated channel):





5. Compute:
$v' \leftarrow (v + \alpha_3)\alpha^{-1} \bmod q$
6. Verify:
$u' \stackrel{?}{=}$
$\text{SHA1}(h', (g^{u'}(h')^{v'} \bmod p)) \bmod q$
7. Keep:
$h', u', v', x_1, \ldots, x_{l-1}, (\alpha^{-1} \bmod q)$

6. Send: v

Some notes on this protocol:

The importance of deleting k: It is vital that the CA never use the same value of k for two different runs of this protocol, and that the CA never reveal k; either case can cause the leaking of the CA's private key. For this reason, k should be securely destroyed immediately after the CA's step 5.

The verification relation: If Victor wants to verify that (u', v') is a valid signature on h', he simply does the same verification that Peggy does in her step 6, namely:

$$u' \stackrel{?}{=} \text{SHA1}(h', (g^{u'}(h')^{v'} \bmod p)) \bmod q$$

Remember, that Peggy must also prove to Victor that she knows a DL-representation of h', or a related value.

5 Showing Protocol

Once Peggy has successfully executed the issuing protocol with the CA, she has a credential with public part (h', u', v') and secret part $(\alpha, x_1, ..., x_{l-1})$ such that

$$u' = \text{SHA1}(h', (g^{u'}(h')^{v'} \bmod p)) \bmod q$$

and

$$h' = f(\alpha, x_1, ..., x_{l-1}) := (g_1^{x_1} \cdot \ldots \cdot g_{l-1}^{x_{l-1}} \cdot h_0)^{\alpha} \bmod p$$

The role of the showing protocol is for Peggy to:

- convince Victor that she indeed possesses a credential as outlined above,
- reveal none, some, or all of the x_i to Victor (but never α); these x_i represent attributes that have been certified by the CA, and
- produce a digital signature on some message m.

The protocol is straightforward: Victor simply checks the first equation above directly, since (h', u', v') is public. Peggy then executes a proof of knowledge of a particular DL representation, as shown in section 2.3 (which gives the digital signature on m for free). Exactly what the DL-representation is, will be clarified by the following:

Example Suppose $l = 5$ and Peggy knows a DL-representation $(\alpha, x_1, \ldots, x_4)$ of h'. Further, suppose she wishes to reveal that $x_2 = 15$ and $x_3 = 23$. That is, she knows (α, x_1, x_4) such that

$$h' = (g_1^{x_1} g_2^{15} g_3^{23} g_4^{x_4} h_0)^{\alpha} \bmod p$$

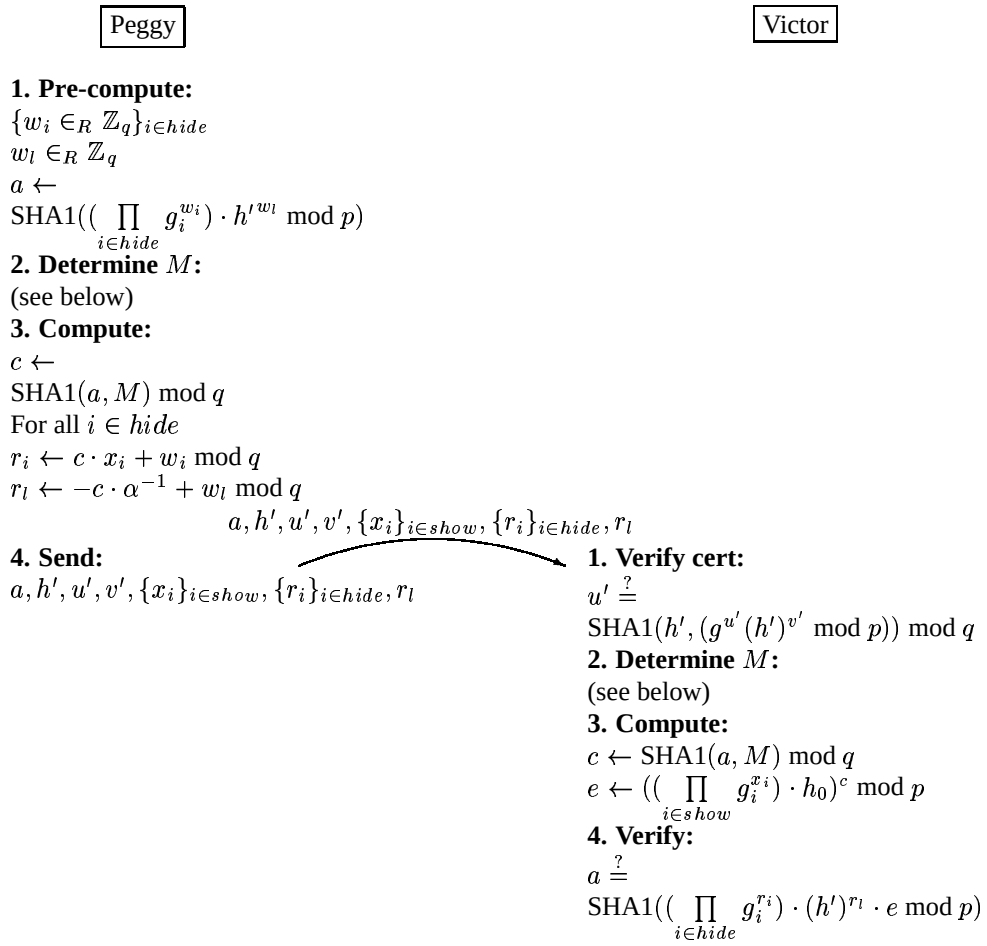
Some rearrangement yields:

$$(g_2^{15} g_3^{23} h_0)^{-1} = g_1^{x_1} g_4^{x_4} h'^{-1/\alpha} \bmod p$$

So all Peggy has to do is use the protocol of section 2.3 to prove that she knows a DL-representation (namely, $(x_1, x_4, -1/\alpha)$) of the number $(g_2^{15} g_3^{23} h_0)^{-1} \bmod p$ with respect to the generators (g_1, g_4, h').

More generally If the set $show$ is the subset of the indices $\{1, 2, \ldots, l-1\}$ corresponding to the attributes which Peggy wishes to reveal to Victor, and $hide$ is the complementary subset, then Peggy is going to prove to Victor that she knows a DL-representation of the number $\left(h_0 \cdot \prod_{i \in show} g_i^{x_i}\right)^{-1} \bmod p$ with respect to the generators $((g_i)_{i \in hide}, h')$.

The protocol In order to prevent replay attacks, Victor must be convinced of the freshness of some nonce. This could be a random value he supplies, a timestamp combined with a unique identifier for Victor, or a value generated by any other method. The nonce will be part of the calculation of c, as was mentioned in section 2.3. This nonce, as well as the message m, the group parameters q and p, the CA's DLREP function f, and the subset $show$ of indices Peggy is willing to reveal to Victor, should be agreed upon by Peggy and Victor before executing the following protocol:



In the above protocol, M is a compound message known to both Peggy and Victor. Victor must be convinced of its freshness, so it should contain the nonce mentioned above. It should also contain the message m on which Peggy desires a digital signature. For extra paranoia, it could also contain other information, such as h', u', v', the attributes Peggy is showing to Victor, p, q, the g_i, the protocol name and version, etc. That is,

$$M := < nonce, m, \text{other information known to both Peggy and Victor} >$$

Exactly which other information is included in M of course also needs to be agreed upon in advance (though the information itself could be part of the message Peggy sends to Victor in her step 4).

6 Limited-show Credentials

It is possible to introduce a security mechanism in the protocols that limits the number of showings of a credential. If the fixed limit, n, is exceeded by Peggy then all attributes of the credential can be computed by a participant having $n + 1$ transcripts of the showing protocol and Peggy loses all privacy. To do so, we limit to n the number of different values of a Peggy can show. Since using the same values of a, and so the same w_i twice, reveals all attributes (see section 2.3), we obtain limited-show credentials.

More precisely, credentials limited to n private showings are achieved by the following simple modifications to the protocols:

- In the issuing protocol: Peggy commits to n different values of a by including them in the credential.
- In the showing protocol: Peggy uses one of the n values for a she committed to in the issuing protocol (instead of creating a new one) and Victor verifies that Peggy's credential specifies n possible values for a and Peggy is indeed using one of them.

For details see [2].

7 Application to e-cash

A simple e-cash system based on one-show private credentials is one with $l = 3$. Each coin consists of:

- A random number $\alpha \in_R \mathbb{Z}_q^*$.
- An identity attribute x_1, which encodes the user's identity or account number. This attribute will never be shown in the showing protocol.
- A value attribute x_2, which encodes the value of the coin, and possibly other public information, such as the currency or expiry date. This attribute will always be shown in the showing protocol.
- The bank's restrictive blind signature (u', v') on $h' = (g_1^{x_1} g_2^{x_2} h_0)^{\alpha}$.

When Peggy asks the bank to issue such a coin with x_1 denoting her identity, and x_2 denoting a certain value, that value is deducted from her bank account. If Peggy wants to spend the coin at Victor's shop, she executes the showing protocol with him, revealing the value of x_2. Victor takes the transcript of this protocol to the bank, which credits him with the value of the coin, as indicated by x_2.

8 Conclusion

We explored in detail the techniques involved in a specific type of credential system which is suitable for use in practice. We discussed limited-show credentials and their application to e-cash systems. The techniques in [2] are flexible and many variations on the protocols exist. We list just a few of those variations and extensions here.

- It is not necessary to work in a subgroup of $\mathbb{Z}_p^*$. Instead, one can use elliptic curves of prime order over certain fields to speed up computations.
- There are also protocols based on the RSA problem, with the nice property that a CA can update Peggy's credential without Peggy having to reveal the private part of it; this is handy for loyalty schemes.
- It is possible not only to show a list of simple attribute values but to show any boolean function of them.
- Limited-show credentials can be used to protect against double-spending in an e-cash system, and verification can be done either on-line or off-line.
- Client-side smart cards can be securely integrated into an e-cash system based on these protocols, so that the card provides the first level of protection against double spending.

References

[1] BRANDS, S. Private credentials. White paper by Zero-Knowledge Systems, Inc., November 2000. Available at: http://www.zks.net/media/credsnew.pdf.

[2] BRANDS, S. *Rethinking Public Key Infrastructures and Digital Certificates*. MIT Press, Cambridge, Massachusetts, August 2000. ISBN 0-262-02491-8.

[3] BRANDS, S. A technical introduction to private credentials. Draft for an upcoming special issue on financial cryptography in the International Journal on Information Security, June 2001.

`$Revision: 1.28 $  $Date: 2001/06/20 19:30:24 $`