International Journal of Sensor Networks and Data Communications

Fan and Adams, Int J Sens Netw Data Commun 2018, 7:4
DOI: 10.4172/2090-4886.1000160

Research Article Open Access

Using Malleable Signatures to Allow Multi-Show Capability in Digital Credentials

Jinnan Fan and Carlisle Adams*

School of Electrical Engineering and Computer Science, University of Ottawa, Ontario, Canada

Abstract

In this paper we propose the use of a malleable signature technique to transform Brands' digital credentials from single-show capability to multi-show capability. Our specific proposed instantiation uses RSA digital signatures so that Alice can efficiently transform an original credential and its corresponding CA signature to a randomized / blinded version of the credential and signature that can be shown to a verifier without risk of traceability (by the CA or across multiple verifiers). We describe our modified issuing and showing protocols and discuss the security properties of our proposed scheme.

Keywords: Digital credentials; Malleable signatures; Privacy technology; Selective show; Multi-show; Cryptography; PKI

Introduction

In [1,2], Brands proposed *digital credentials* as a privacy enhancing technology for end users. A digital credential is a data structure that allows its holder to determine for herself when, how, and to what extent she is willing to reveal her attributes to others, and to what extent others can link or trace this information. This means that users in this system do not need to trust third parties to protect their privacy (in particular, even if all parties in the system have unlimited computing power, they cannot learn more than what users willingly disclose).

Digital credentials involve three parties: users, verifiers, and a CA (Certification Authority). A user's attributes can be considered as her private key; these are then encoded into the corresponding public key (*i.e.*, the digital credential). The digital signature of a trusted CA on the public key enables users to selectively disclose their specific attributes or even some properties of the attribute values while keeping the remaining attributes completely hidden from other parties, including parties in the transaction and any eavesdroppers that listen to the communication channel [3].

Background and related work

Digital credentials technology is based on fundamental concepts proposed by Chaum [4], including blind signatures [5,6], untraceable electronic cash [7], group signature schemes [8], pseudonym credential systems [9,10] and one-show blinding [11]. It also builds on work in commitment schemes by several researchers (see, for example, Brassard et al. [12], Chaum and Van Antwerpen [13], Chaum [14], Pedersen [15,16], and Van Heyst and Pedersen [17], etc.). Since [1,2], a number of variations and alternative credentials schemes have been published, particularly those based on the unlinkable anonymous credentials of Camenisch and Lysyanskaya (see [18] for the original proposal).

Anonymous credentials [18] construct a pseudonym of the user for use with a specific organization. Each pseudonym is tagged with a value, called a *validating tag*. Proof of possession of a credential is achieved through a statistical zero-knowledge proof of knowledge of a correctly-formed validating tag and its corresponding credential. The actual credential is never revealed in a showing protocol (which is why it is multi-show); thus, Camenisch and Lysyanskaya's credential system is quite different from Brands' digital credential system [1,2]. But zero-knowledge proofs can involve significant computation and may not be ideally suited to all situations and environments. In this paper we propose a simple mechanism to make Brands' credentials multi-show.

Research motivation and goals

This research has the goal of solving the problem of linkability of Brands' digital credentials [1]. Thus, we seek to create a protocol that achieves the following:

1. Every transaction Alice makes will not be traced back to her.
2. Any two transactions Alice makes will not be linked to each other.

In line with cryptographic tradition and Brands' digital credential model, we will use fictitious characters Alice as a digital credential holder and also the user in our system, and Bob as a digital credential verifier. The CA will directly issue digital credentials to Alice.

Existing Blind Signatures

Brands' restrictive blinding

Brands proposed a technique called *restrictive blinding* in order to hide the digital credential from the CA, so that the CA cannot later trace Alice's transactions [1]. Since Alice needs to reveal her digital credential and the corresponding signature issued by the CA when disclosing any of her attributes to others, the CA should not see either the credential or the signature when issuing the credential (*i.e.*, the CA needs to blindly sign Alice's digital credential). Using *restrictive blind signatures*, Brands proposed a basic issuing protocol as follows.

Figure 1 shows the basic issuing protocol from Brands' [1]. In this protocol, the tuple $(x_1,..., x_l)$ is Alice's private key (in particular, these correspond to her attributes) for the digital credential, and she will

***Corresponding author:** Carlisle Adams, School of Electrical Engineering and Computer Science, University of Ottawa, Ontario, Canada, E-mail: jfan084@uottawa.ca; cadams@uottawa.ca

Received November 10, 2018; **Accepted** November 29, 2018; **Published** December 07, 2018

Citation: Fan J, Adams C (2018) Using Malleable Signatures to Allow Multi-Show Capability in Digital Credentials. Int J Sens Netw Data Commun 7: 160. doi: 10.4172/2090-4886.1000160

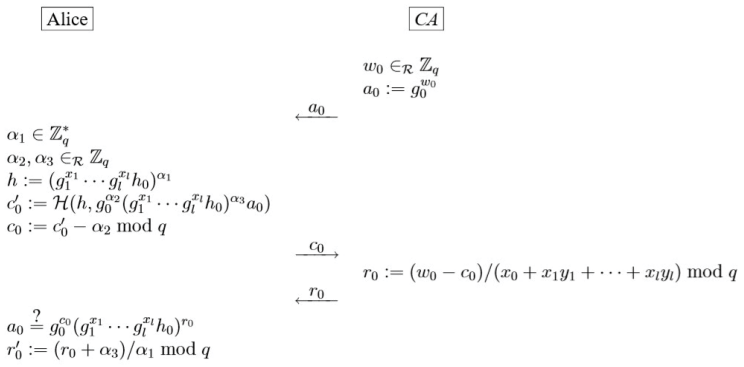

Figure 1: Brands' basic issuing protocol (from [5]).

obtain the CA's signature on the corresponding public key h. α_1 is a credential blinding factor randomly chosen and kept secret by Alice. α_2 and α_3 are also randomly chosen and kept secret by Alice and are used in the subsequent signature blinding computation. H(·) is a strong one-way hash function. As shown in the protocol, the CA generates a random number w_0 and then constructs the challenge a_0. Note that (g_0,..., g_l, h_0) are system parameters (known by all parties) and (y_1,..., y_l, x_0) are private values known only by the CA.

After Alice and the CA complete this issuing protocol, Alice will hold the digital credential public key h and CA signature (c_0', r_0'), which she can then show to Bob. In the meantime, all that the CA sees in this issuing process is (a_0, c_0, r_0), which are different from what Alice is going to disclose in a showing protocol to verifiers.

From the above Figure 1, we can see that the CA must also know the value of Alice's attributes (x_1,··· ,x_l) in this basic issuing protocol. To address this problem, Brands proposed another issuing protocol with attribute hiding, which allows the CA to recertify previously issued digital credentials and then to issue new ones without knowing the attributes they contain (for environments where this level of privacy may be important).

Limitations

In a general digital credential system, let's say Alice has her own digital credential h, which contains her attributes, *e.g.*, age (25), credit card number (x)...etc. In order to make her credential valid, she needs to get the CA's signature on her credential, *i.e.*, *sig* (h). Then Alice wants to buy alcohol from Bob, so she needs to prove that she is of legal age and thus she shows 25, h and *sig* (h) to Bob. If she also wants to buy a new laptop from David, she will show x, h and *sig* (h) to David.

From this simple example of credential use, we can see that at least two kinds of attacks can be used to track Alice:

1. The CA can collude with Bob so that Bob learns Alice's credit card number, and/or the CA can collude with David so that David learns Alice's age.

2. Bob and David can collude directly so that they each learn Alice's age and credit card number.

In Brands' proposed digital credential system, the first attack is mitigated through the use of blind signatures: the CA does not see Alice's actual credential and signature; thus, the CA cannot recognize when she uses it with Bob (or David) and so cannot collude with them. On the other hand, the second attack cannot be avoided. Alice needs to show her credential h when she needs to reveal any of the attributes contained in h, even though she may reveal different attributes to different verifiers. In this case, Bob/David would learn that all these actions are performed by the same individual and all the transactions involving this specific digital credential are linked.

Note that if Alice discloses attributes or combinations of attributes that are unique to her (e.g., her passport number, or her name and home address), then she can be tracked regardless of the protections put in place in the issuing and showing protocols themselves. This is not a problem that can be solved through technical means and so we do not address this situation in this paper. (In general, Alice needs to be careful about what attributes she reveals to which parties if she is concerned about protecting her privacy).

In conclusion, digital credentials created by applying Brands' techniques can only be used once because if they are used more than that, the user's transactions will be traceable between different verifiers.

Malleable signatures

In Brands' original proposal, the blinding process happens once (i.e., between Alice and the CA during the issuing protocol, so that the CA cannot later trace Alice's movements as she uses her credential and signature). However, once Alice has her issued credential and signature, she would use these with all verifiers. Her transactions can therefore be linked across different verifiers; furthermore, collusion among verifiers is possible so that each of them can learn more about Alice. Our proposal is to have the blinding process happen in the showing protocol with every transaction. In this way, different verifiers will not know that they have interacted with the same entity (i.e., Alice) and so linking of her transactions and collusion among verifiers are both prevented. (Note that it also remains true that the CA cannot trace Alice's movements because Alice is using a randomized credential every time).

In order to have the blinding process in the showing protocol, it is necessary for Alice to not only randomize the credential, but to correspondingly randomize the CA's signature so that this "new" credential can be verified. In other words, Alice requires a "new" CA signature, unlinkable to the original signature, which can verify the "new" credential using the CA's public key, but of course without requiring the CA's private key to create the "new" signature.

To address this counterintuitive problem, we build on the concept of malleable signatures as proposed and discussed in a growing body

of research papers (see, for example, definitions and delegatable anonymous credentials [19-21], implementations, sanitizable, and redactable signatures [22-27], bounded vector signatures [28], and delegatable functional signatures [29]).

Generally, a signature scheme is malleable if, for a given message and its signature, it is possible to efficiently modify the signature to be a valid signature on a related massage without using the private signing key [21].

Ateniese et al. [22] presented the notion of sanitizable signatures, which allow another party to modify designated portions of a document and then produce a valid signature on the modified document without help from the signer. Brzuska et al. [23] then constructed a sanitizable signature scheme with perfect unlinkability between sanitized message-signature pairs of the same document. Fleischhacker et al. [24] then developed this by re-randomizing the message signing and verification keys. Ma et al. [26] presented an efficient construction of authenticated data redaction with fine-grained redaction control.

Wei et al. [28] proposed *bounded vector signatures* which allow a user to increase the value embedded in any component of the signed vectors to a pre-defined bound without access to the signing key. Backes et al. [29] introduced *delegatable functional signatures* which allow the signer to delegate the signing capability to another party and also specify how this party can modify the signature or further delegate its capability.

Chase et al. [20,21] gave new (extended) definitions of *malleable signature* and *malleable zero-knowledge proof*, which allow them to construct malleable signatures with a wider range of transformation classes and then construct delegatable anonymous credentials from these signatures. Blömer et al. [19] identified a new primitive called *dynamically malleable signatures*, in which the set of allowed transformations is not static but can be changed over time for each signature.

Practical implementations of malleable signature schemes have also been demonstrated by various researchers (see, for example, [25,27]).

Note that all these previous papers define and explore different types (instantiations) of malleable signatures but none of them are affiliated with Brands' digital credential scheme and, specifically, none have been used to make these credentials multi-show.

The challenge in our work is to find an efficient malleable signature construction for Brands' credentials that allows the signature to be blinded / randomized in such a way that

1. It is provably unlinkable to the original signature,
2. It can be verified using the original (i.e., the CA's) public key, and
3. It shows integrity and authenticity of only a specific piece of data (i.e., the corresponding randomized credential). In particular, Alice must not be able to use malleability to construct a valid-looking CA signature on data that the CA would not have signed; it must be a signature on a randomized credential that contains all and only her original attributes.

Our construction uses the malleability of the RSA digital signature scheme. We demonstrate our proposal as four parts: **Initialization**, **Issuing** protocol, **Showing** protocol, and **Verification**.

There are basically two steps that need to be done in **Initialization**.

1. The CA generates appropriate system parameters and makes them available to all users in the system.
2. The CA generates its key pair for credential signing and verification purposes.

The main steps of our proposed **Issuing** protocol are as follows:

1. Alice has her initial digital credential h, which contains all her attributes, such as her age (20), credit card number (x)...etc.
2. The CA then adds its serial number (a fixed value) as the last attribute of Alice's initial credential.
3. Alice and the CA communicate with each other to get w via a Diffie-Hellman (D-H) key exchange. (If they fully trust each other, it is not necessary to do D-H key exchange to obtain w; rather, this parameter could be generated by either of them and simply given to the other.)
4. The CA signs both h and w to get sig_h and sig_w, respectively. The CA then sends these signatures to Alice.

Note that either Alice or the CA could create Alice's credential h and show it to the other. In practice, they need to meet in person to ensure that these attributes do in fact belong to Alice and this credential has not been comprised. In other words, the integrity of the credential must be guaranteed but confidentiality between Alice and the CA is typically less important. Note that if attributes need to be added, deleted, or modified, a new credential will need to be issued by the CA. This is identical to the credentials that we use in the physical world today. For example, we can view the personal information presented on a driver's license as attributes. Any data (e.g., the home address) in an existing license that needs to be modified will require a new driver's license to be issued.

The main steps of our proposed **Showing** protocol are as follows:

1. Alice uses her initial credential, her parameter w, and the values sig_h and sig_w to generate a one-time showing credential h_{show} and corresponding signature sig_{show}.
2. Alice sends the value(s) of the attribute(s) she wants to show, the CA's serial number, h_{show}, and sig_{show} to the verifier. (We assume that the position of the attributes in the credential (i.e., the position i of each attribute) will be standardized and known to all verifiers.)
3. If Alice wants to show any attributes contained in the credential to other verifiers, she needs to repeat Steps 1 and 2.

The main steps of proposed **Verification** are as follows:

1. Bob or other verifiers obtain the verification key (public key) and the serial number from the appropriate CA.
2. Verify if the one-time showing credential and signature are valid or not.
3. Verify if the disclosed attribute(s) is (are) valid or not.

As with Brands' original proposal, our system has three related parties: the CA, users, and verifiers. The CA has an RSA key pair for credential signature purposes (with private signing exponent d and public verification exponent e); it issues to a user a signed digital credential h. The user, Alice, uses her initial credential h to create a one-time showing credential and corresponding one-time showing signature, which will be communicated to a verifier. A verifier, Bob

or David, will get the verification public key from the appropriate CA and use it to verify whether the one-time showing credential and corresponding signature are valid or not. He will then verify if the user's disclosed attributes are valid or not.

Our complete proposal, consisting of a modified issuing protocol and a modified showing protocol, is described in sections 3.1-3.3, followed by a high-level security analysis.

Issuing process

Figure 2 shows our proposed issuing protocol. In this protocol, the tuple (x_1,..., x_{l-1}) is Alice's private key (her attributes) for her digital credential h. She chooses her secret value α and will never reveal it. The CA's private (signing) key is d and its corresponding public (verification) key (used by Bob in Figure 3) is e.

Selective disclosure multi-Show process

Figure 3 presents our proposed selective disclosure multi-show protocol. Bob chooses a random number a and then sends this challenge to Alice. Alice chooses her secret value b and then uses (a, b) to construct her one-time showing credential h_{show} and signature sig_{show}. In the example shown in Figure 3, Alice chooses to disclose x_1 while keeping her other attributes hidden. She constructs (r_0, r_1,..., r_l) using a, b, her secret value α, her remaining hidden attributes x_2, …, x_{l-1}, and the w_i from Figure 2.

Authority key pair

We use the basic RSA digital signature algorithm for the CA key pair in our malleable signature construction, which means there is no padding or hashing in the signing and verification processes.

Security analysis

Note that due to space limitations, this paper contains a brief outline of the security properties of our proposed protocol. A forthcoming paper will give a more detailed security analysis, along with specifics

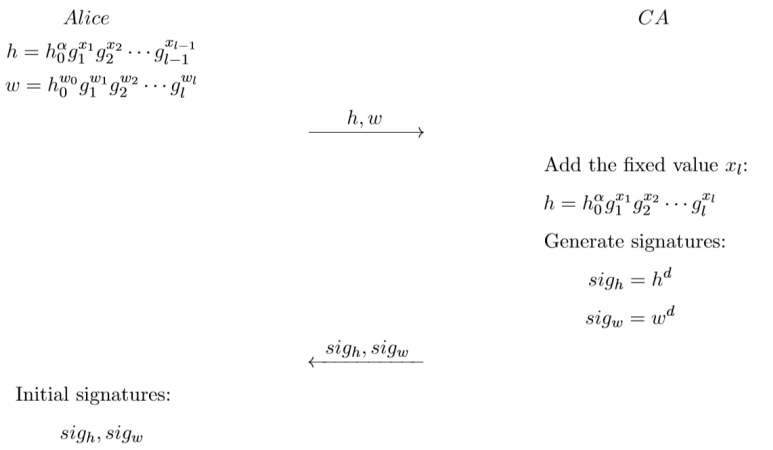

Figure 2: Issuing protocol.

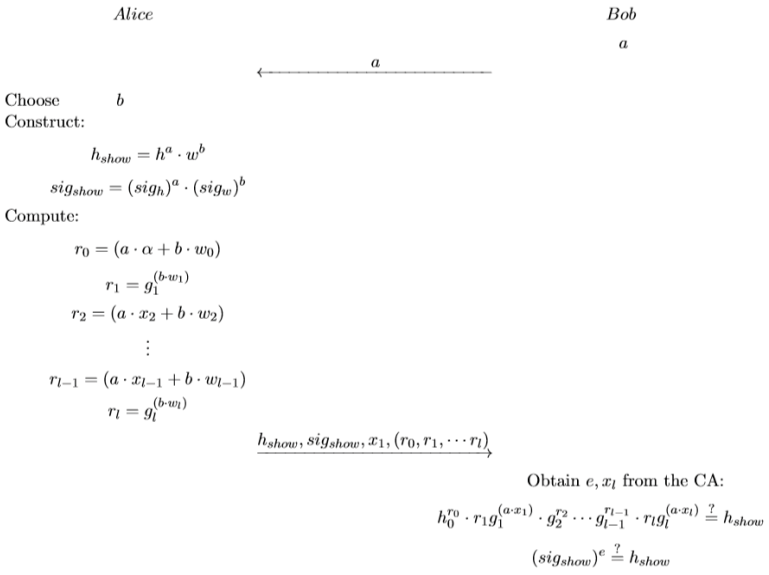

Figure 3: Showing protocol.

of the construction (including constraints on parameter values, sizes of all parameters to achieve a given security level, and creation of an appropriate modulus for all computations).

Regarding the two kinds of tracing problems mentioned in Section 2.2, our proposed protocol has mitigated both attacks. Alice will never show her initial credential h and signature sig_h to any verifiers since she will modify them with every showing execution. Thus, the CA cannot learn which credentials refer to the Alice, and there is no way for Bob, David or the CA to collude to trace the transactions Alice makes because she never shows the same credential & signature more than once. (Note that Alice's original credential and signature are used as the basis for every new showing transaction; in this sense they are multi-show, but each randomized credential & signature pair is used only in a single showing transaction. Thus, Alice derives single-show values for each transaction whenever she wishes, rather than obtaining a single-show value from the CA at issuing time.)

Secret parameters: There are two parameters that Alice chooses/generates herself and never shares with anyone:

1. b from $h_{show} = h^a.w^b$: this keeps others from being able to link the one-time showing credential and Alice's initial credential. We will discuss the detailed functionality of b to Alice in Section 3.4.2.

   A new random value for b will be chosen with every showing process.

2. α from $h = h_0^{\alpha}.g_1^{x_1}.g_2^{x_2}\ldots$: this may be randomly chosen by Alice, or may be derived from Alice's biometric or her passphrase. If Alice is the only entity that knows α then no other entity can prove knowledge/possession of the attributes contained in h (this property was proven in Brands' original scheme).

Functionality of *a* and *b*: When Alice creates h_{show} and sig_{show} during the showing protocol, she needs to communicate with Bob about the parameter a.

Functionality of *a*: The randomness from Bob prevents Alice from forging signatures, particularly sig_{show}.

Functionality of *b*: The randomness from Alice prevents others from relating h_{show} and sig_{show} to h and sig_h.

Specifically, from the structure of $h_{show} = h^a.w^b$ and $sig_{show} = (sig_h)^a.(sig_w)^b$, we can see that

- Bob knows the value of a, h_{show}, and sig_{show}.
- The CA knows the value of h, sig_h, w, and sig_w. (Note that, from the issuing protocol, the CA knows the h and w that belong to Alice.)

If either Bob or the CA learns the value of b (or, in the degenerate case, if $b = 1$), they can collude to know the specific h_{show} and sig_{show} that correspond to the initial h and sig_h, which means that Alice's one-time showing credential and signature will be learned and traced back to the initial credential and signature issued by the CA.

Functionality of the fixed attribute: Adding an attribute with fixed value to Alice's initial credential is necessary to prevent Alice from illegally manipulating her issued credential.

As we presented above, Alice's initial issued credential is as follows:

$$h = h_0^{\alpha} g_1^{x_1} g_2^{x_2} \ldots g_l^{x_l}$$

Suppose that Alice raises h to the power m:

$$h^m = h_0^{\alpha m} g_1^{x_1 m} g_2^{x_2 m} \ldots g_l^{x_l m}$$

Now, Alice has a new valid-looking initial credential:

$$h' = h_0^{\alpha'} g_1^{x_1'} g_2^{x_2'} \ldots g_l^{x_l'},$$

where $h' = h^m, \alpha' = \alpha m, x_1' = x_1 m, x_2' = x_2 m, \ldots x_l' = x_l m$.

She can now compute a corresponding signature as follows:

$$sig_h' = sig_h^m = (h^m)^d = (h_0^{\alpha m} g_1^{x_1 m} g_2^{x_2 m} \ldots g_l^{x_l m})^d = (h_0^{\alpha'} g_1^{x_1'} g_2^{x_2'} \ldots g_l^{x_l'})^d$$

Alice can obtain any value of the attributes by manipulating the value of m appropriately. The corresponding sig_h' would appear to be perfectly valid to a verifier. The fake pair of initial credential and signature, h' and sig'_h, can then be easily modified as a one-time showing pair, h'_{show} and sig'_{show}, by following the structure in Section 3.2.

To avoid this attack, the value of x_l is fixed (it is the CA's serial number). In this case, h'_{show} would be immediately detected as being compromised at verification time. To be specific, there are three possible situations during the selective disclosure process:

1. Alice discloses the values of real attribute x_1 and real serial number x_l. The verification,

   $$h_0^{r_0}.r_1.g_1^{(a.x_1)}.g_2^{r_2}\ldots r_l.g_l^{(a.x_l)} = h_{show},$$

   means that the disclosed attributes are valid (as demonstrated in Section 3.2).

2. Alice discloses the values of fake attribute x_1' and fake serial number x_l'. The verification,

   $$h_0^{r_0}.r_1.g_1^{(a.x_1')}.g_2^{r_2}\ldots r_l.g_l^{(a.x_l')} = h'_{show},$$

   means that this fake credential h'_{show} and the disclosed attributes appear valid. But it is easy for Bob to see that the serial number is fake (because Bob knows the real serial number of the CA); thus, x_1' is fake as well.

3. Alice discloses the values of fake attribute x_1' and real serial number x_l. The verification,

   $$h_0^{r_0}.r_1.g_1^{(a.x_1')}.g_2^{r_2}\ldots r_l.g_l^{(a.x_l)} \neq h'_{show},$$

   means that the disclosed attributes are invalid.

   If Alice just replaces the real attribute x_1 by x_1' and lets the remaining attributes (including x_l) have the original values in h_{show}, then the verification,

   $$h_0^{r_0}.r_1.g_1^{(a.x_1')}.g_2^{r_2}\ldots r_l.g_l^{(a.x_l)} \neq h_{show},$$

   means that the disclosed attributes are invalid.

Results and Discussion

The use of malleable signatures from the CA in Brands' digital credentials allows the blinding process to be moved from the issuing protocol (i.e., Alice's interaction with the CA during the creation of the credential) to the showing protocols (i.e., Alice's interaction with a verifier during the use of the credential). This proposal, and its instantiation using the malleability of the RSA digital signature scheme, provides an efficient way to turn Brands' credentials from single-show to multi-show capability without sacrificing any of their original security or privacy properties. With randomization of the

credential and its associated signature in the showing protocol, Alice is able to prevent the linking of her transactions (both to her and to other transactions) by the CA as well as by all verifiers.

With respect to further work, we are currently pursuing two directions.

- We are examining how to efficiently combine our protocol with additional biometric techniques to make our multi-show digital credentials non-transferable. This will ensure that Alice cannot lend her credential to several friends so that, for example, they all enjoy access to a subscription service for the price of a single user.
- We are also looking at effective ways to extend our protocol to avoid replay attacks. In particular, we are considering the best places and ways to add nonces or timestamps, for example, so that credentials cannot be maliciously replayed by any party.

References

1. http://www.cypherspace.org/credlib/brands-technical.pdf
2. Brands S (2000) Rethinking public key infrastructures and digital certificates: building in privacy. MIT Press.
3. Adams C (2011) Achieving non-transferability in credential systems using hidden biometrics. Security and Communication Networks 4: 195-206.
5. Chaum DL (1981) Untraceable electronic mail, return addresses, and digital pseudonyms. Commun ACM 24: 84-90.
5. Chaum D (1983) Blind signatures for untraceable payments. Advances in cryptology. Springer, Boston, MA, Pp: 199-203.
6. Chaum D (1984) Blind signature system. Advances in cryptology. Springer, Boston, MA, Pp: 153.
7. Chaum D, Fiat A, Naor M (1988) Untraceable electronic cash. Conference on the Theory and Application of Cryptography. Springer, New York, NY, Pp: 319-327.
8. Chaum D, Heyst EV (1991) Group signatures. Workshop on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, Pp: 257-265.
9. Chaum D (1985) Security without identification: Transaction systems to make big brother obsolete. Commun ACM 28: 1030-1044.
10. Chaum D, Evertse JH (1986) A secure and privacy-protecting protocol for transmitting personal information between organizations. Conference on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, Pp: 118-167.
11. Chaum D (1990) One-show blind signature systems. U.S. Patent No. 4,914,698. 3 Apr. 1990.
12. Brassard G, Chaum D, Crépeau C (1988) Minimum disclosure proofs of knowledge. J Comput Syst Sci 37: 156-189.
13. Chaum D, Antwerpen HV (1989) Undeniable signatures. Conference on the Theory and Application of Cryptology. Springer, New York, NY, Pp: 212-216.
14. Chaum D (1990) Zero-knowledge undeniable signatures. Workshop on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, Pp: 458-464.
15. Pedersen TP (1991) Non-interactive and information-theoretic secure verifiable secret sharing." Annual International Cryptology Conference. Springer, Berlin, Heidelberg, Pp: 129-140.
16. Pedersen TP (1992) Distributed provers and verifiable secret sharing based on the discrete logarithm problem. DAIMI Report Series 21.
17. Heyst EV, Pedersen TP (1992) How to make efficient fail-stop signatures. Workshop on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg Pp: 366-377.
18. Camenisch J, Lysyanskaya A (2001) An efficient system for non-transferable anonymous credentials with optional anonymity revocation. International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, Pp: 93-118.
19. Blömer J, Bobolz J (2018) Delegatable Attribute-Based Anonymous Credentials from Dynamically Malleable Signatures. International Conference on Applied Cryptography and Network Security. Springer, Pp: 221-239.
20. Chase M, Kohlweiss M, Lysyanskaya A (2013) Malleable Signatures: Complex Unary Transformations and Delegatable Anonymous Credentials. IACR Cryptology ePrint Archive 2013: 179.
21. Chase M, Kohlweiss M, Lysyanskaya A, Meiklejohn S (2014) Malleable signatures: New definitions and delegatable anonymous credentials. Computer Security Foundations Symposium (CSF), 2014 IEEE 27th. IEEE 2014.
22. Ateniese G, Daniel HC, Breno de M, Tsudik G (2005) Sanitizable signatures. European Symposium on Research in Computer Security. Springer, Berlin, Heidelberg.
23. Brzuska C, Pöhls HC, Samelin K (2013) Efficient and perfectly unlinkable sanitizable signatures without group signatures. European Public Key Infrastructure Workshop. Springer, Berlin, Heidelberg, Pp: 12-30.
24. Fleischhacker N, Krupp J, Malavolta G, Schneider J, Schröder D, et al. (2018) Efficient unlinkable sanitizable signatures from signatures with re-randomizable keys. Iet Inform Secur 12: 166-183.
25. Lenz T, Krnjic V (2018) Towards Domain-Specific and Privacy-Preserving Qualified eID in a User-Centric Identity Model." 17th IEEE International Conference on Trust, Security and Privacy in Computing And Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE). IEEE, 2018.
26. Ma J, Liu J, Huang X, Xiang Y, Wu W, et al. (2017) Authenticated data redaction with fine-grained control. IEEE Transactions on Emerging Topics in Computing: 1.
27. Pöhls HC, Peters S, Samelin K, Posegga J, Meer HD, et al (2013) Malleable signatures for resource constrained platforms." IFIP International Workshop on Information Security Theory and Practices. Springer, Berlin, Heidelberg, Pp: 18-33.
28. Wei L, Coull SE, Reiter MK (2011) Bounded vector signatures and their applications." Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security. ACM 2011.
29. Backes M, Meiser S, Schröder D (2016) Delegatable functional signatures. Public-Key Cryptography–PKC 2016. Springer, Berlin, Heidelberg, Pp: 357-386.