An Application of a Group Signature Scheme with Backward Unlinkability to Biometric Identity Management

Julien Bringer[1], Hervé Chabanne[1,2] and Alain Patey[1,2]
[1]*Morpho, Issy-Les-Moulineaux, France*
[2]*Télécom ParisTech, Identity and Security Alliance (The Morpho and Télécom ParisTech Research Center), Paris, France*

Keywords: Group Signatures, Identity Management, Derivation, Cascade Revocation, Biometrics, Anonymity.

Abstract: We introduce a new identity management process in a setting where users' identities are credentials for anonymous authentications. Considering identity domains organized in a tree structure, where applying for a new identity requires to previously own the parent identity, we enable a cascade revocation process that takes into account this structure while ensuring anonymity for non-revoked users, in particular, towards the providers of other identity domains. Our construction is based on the group signature scheme of (Bringer and Patey, 2012).

1 INTRODUCTION

In this paper we consider a scenario where users have access to a kind of federation of identity management systems with different identity providers that have some dependencies between them: To each identity provider corresponds an identity domain and the set $\mathbb{I}$ of these domains is structured as a tree. When one wants to apply for a new identity in an identity domain I_l, one has to own a valid identity for the parent domain I_k. These dependencies also imply that it should be possible to automatically revoke across different domains. To this aim, the new identity is derived from the previous ones in order to maintain a link with the identities above. Contrary to what is done in centralized federated identity management, one important issue is then to ensure the privacy of this link. We call this property *Cross-Unlinkability*

Let us give an example of application of our proposal. Consider the identity domain (sub-)tree described in Figure 1. We assume that a government sets up an identity management system, used for instance to access services. In this example, applying for an identity stating that you own a car insurance requires to previously own an identity in the domain of users with driver's licenses. We also wish that, when a user uses his student identity, anonymity of this user is guaranteed against the providers of all other domains, including the managers of the parent domain (National Identity), the children domains (Colleges) or the sibling domains (Driver's license).

We use as elementary component of our system a

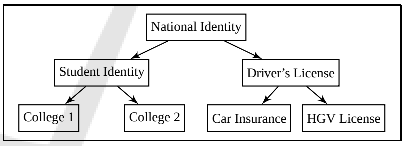

Figure 1: An example of an identity domain tree $\mathbb{I}$.

biometric anonymous authentication scheme (Bringer et al., 2008) (BCPZ) based on Verifier-Local Revocation (VLR) group signatures (see Figure 2). This protocol enables members of a group, managed by a Group Manager GM, to authenticate, using an electronic device, to a service provider while proving nothing more than their belonging to the group. The use of biometrics guarantees that the legitimate user uses the device and this, combined to the use of a group signature scheme, leads to an anonymous remote authentication. We can see the group considered in such a scheme as an identity domain I_l, where the identity provider IP is the GM. The keys that are issued by IP are actually credentials that are associated to the issued identity. In the following, these credentials will be assimilated to the identities. The users that obtained an identity from IP can prove its validity to service providers that rely on this identity domain.

We recall that group signatures enable authorized users to sign anonymously on behalf of a group. We only consider the case of VLR group signatures. The VLR property (Boneh and Shacham, 2004) guarantees that only the public parameters and a revocation list RL are required to check a signature. Concretely, when a user is revoked, a revocation token derived

Bringer J., Chabanne H. and Patey A..
An Application of a Group Signature Scheme with Backward Unlinkability to Biometric Identity Management.
DOI: 10.5220/0004123904210425
In *Proceedings of the International Conference on Security and Cryptography* (SECRYPT-2012), pages 421-425
ISBN: 978-989-8565-24-2

from his signing key is added to RL. It is used by verifiers to prevent revoked users from further signing.

To reach our goal of Cross-Unlinkability, we use the group signature introduced in (Bringer and Patey, 2012) (which patches and extends (Chen and Li, 2010)) that satisfies *Backward Unlinkability*. This property enables users to sign at different time periods using the same keys, while maintaining unlinkability between signatures issued at different periods, even if the user is revoked at one of these periods. In our proposal, we no more consider these periods as time periods but as children of a given identity in the identity domain tree. Thus, authentications in two different domains are impossible to link if the user is not revoked from both. Moreover, the cascade revocation process that we describe does not threaten the security properties that we guarantee.

2 THE CL AND BP GROUP SIGNATURES

In this section, we describe the model of group signatures presented in (Bringer and Patey, 2012). We instantiate this model using two schemes introduced in (Bringer and Patey, 2012): a patched version of the (Chen and Li, 2010) scheme, denoted by CL, and an extension of this patched version with Backward Unlinkability (BU), denoted by BP. Notice that both can be used with the same parameters.

2.1 Components

There are three types of entities: a Group Manager GM, a set of members and a set of verifiers. A BP or a CL Group Signature Scheme consists of the following algorithms. (Moreover, in the BP scheme, because of BU, all algorithms but **KeyGen** depend on the current time period j and one revocation list RL_j per time period has to be used (see also Remark 1)).

KeyGen. The group manager outputs the group public parameters gpk. He also chooses a secret key msk and its public counterpart mpk. gpk and mpk are published. GM also publishes an empty revocation list RL.

Join. This algorithm is an interactive protocol between GM and a member M_i. M_i gets a secret key $sk_i = (x_i, A_i, f_i)$ where f_i is chosen by M_i, x_i by GM and A_i is computed by GM using msk, x_i and some information about f_i. GM only gets x_i and A_i, he also derives a revocation token rt_i from x_i.

Revoke. GM runs this algorithm to prevent a member M_i from further making valid signatures. It outputs an updated revocation list RL.

Sign. This algorithm, run by a member M_i, takes as input a message m, M_i's key sk_i and a message m. It outputs a signature σ.

Verify. This algorithm, run by a verifier takes as input a message m, its signature σ and the Revocation List RL. It checks if the message has been signed by an unrevoked group member, without revealing the signer's identity. The possible outputs are valid and invalid.

Open. This algorithm is run by GM. It takes a signature σ on a message m as input, together with all revocation tokens of the group members. It reveals the identity of the signer.

2.2 Security Properties

We describe the security properties fulfilled by the group signature schemes. Both BP and CL schemes satisfy *Correctness*, *Selfless-Anonymity*, *Traceability* and *Exculpability*. The BP scheme moreover satisfies *Backward Unlinkability*.

(a) **Correctness.** Every check of a well-formed signature, made by an unrevoked user, returns valid.
(b) **Selfless-Anonymity.** A member can say if he produced a particular signature. If it was not him, he has no information about the user who produced it.
(c) **Traceability.** No attacker (or group of attackers) is able to forge a signature that can not be traced to one of the corrupted users which participated in its forgery.
(d) **Exculpability.** Nobody, even the Group Manager, is able to produce another user's signature.
(e) **Backward Unlinkabilty.** (encompasses *Selfless-Anonymity*) The valid signatures remain anonymous, even after the signer's revocation. Revoked users can come back after their revocation into the group and use their previous keys without any loss of anonymity.

Remark 1. *(Backward Unlinkability) To enable BU, the BP scheme divides time into periods. Instead of a unique revocation list RL, there is one revocation list RL_j for each period j. Similarly, each member M_i has a revocation token rt_{ij} for each period j instead of a unique rt_i. Usually, for every time period j, a random token h_j is chosen. The period revocation token is then obtained as follows: $rt_{ij} = h_j^{rt_i}$. Thus, two tokens rt_{ij} and $rt_{ij'}$ of the same user at different time periods are unlinkable, which guarantees BU.*

Remark 2 (BCPZ Anonymous Authentication)**.** *We describe in Figure 2 how to adapt the BCPZ anonymous authentication scheme using the CL scheme. We refer the reader to (Bringer et al., 2008) for further details. Notice that in our adaptation, we use the*

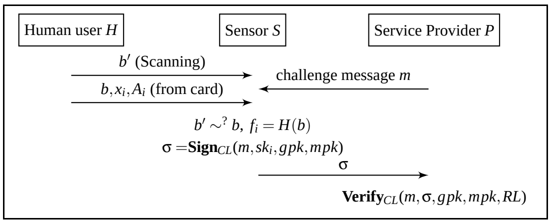

Figure 2: The BCPZ authentication scheme.

property of Exculpability enabled by the CL scheme and we do not give any biometric data to the GM.

3 OUR PROPOSAL

3.1 The Model

We assume that identity domains are organized as a tree $\mathbb{I}$ with a root $\mathcal{I}_0$. When one wants to acquire a new identity from a domain $\mathcal{I}_l$, one has to prove that one owns a valid identity for its parent domain $\mathcal{I}_k$ in $\mathbb{I}$. Each identity domain $\mathcal{I}_l$ has an identity provider IP_l and we will denote by $k \prec l$ the fact that the identity domain $\mathcal{I}_k$ is the parent of the identity domain $\mathcal{I}_l$.

For each identity that they own, the authorized users possess the necessary keys to authenticate anonymously following the principle of the BCPZ scheme. The corresponding IP is in fact the group manager for the underlying group signature scheme. The functionalities of our protocol are the following.

KeyGen. This is run by the IP's. IP_0 first returns the public parameters gpk for all the domains. Then each IP_l creates a secret/public key pair (msk^l, mpk^l) and publishes mpk^l.

Enrolment. For a domain $\mathcal{I}_l$, this algorithm is jointly run by the identity provider IP_l and a user M_i. The input for the user is a fresh acquisition b_i of a biometric trait B and for IP_l is his secret key msk^l. It returns a new secret key $sk_i^l = (x_i^l, A_i^l, f_i^l)$ for M_i for the identity domain $\mathcal{I}_l$. f_i^l is only known from M_i and the other parts x_i^l and A_i^l are known from both. x_i^l is in particular used as a revocation token rt_i^l for M_i for this domain.

Derivation. For a domain $\mathcal{I}_l$, this algorithm is jointly run by a user M_i requiring to get a new identity for the domain $\mathcal{I}_l$ and the identity provider IP_l of $\mathcal{I}_l$. The input for M_i is his secret key for the parent domain $\mathcal{I}_k$ of $\mathcal{I}_l$ in $\mathbb{I}$ and the input for IP_l is his secret key msk^l. It returns the result of the enrolment of M_i to $\mathcal{I}_l$ if M_i successfully proves to IP_l that he owns a valid (and non revoked) identity for $\mathcal{I}_k$.

Authentication. For a domain $\mathcal{I}_l$, this is jointly run by a user M_i and a service provider requiring a valid identity from $\mathcal{I}_l$. The input of M_i is his secret sk_i^l and a fresh biometric acquisition b_i'. The service provider only needs gpk and mpk^l. It returns a boolean denoting the acceptance or the reject of the authentication.

Revocation. This recursive algorithm is run by the identity provider IP_l of $\mathcal{I}_l$ who wants to revoke a member M_i of $\mathcal{I}_l$. It takes as input the revocation token rt_i^l of the user M_i and the revocation list RL_l.

- Local Revocation: It returns an updated RL_l where the revocation token of M_i for $\mathcal{I}_l$ is added.
- Downwards Revocation (compulsory): The newly published revocation token rt_i^l is sent to the IP's of the domains that are children of $\mathcal{I}_l$, who then run the *Revocation* algorithm.
- Upwards Revocation (optional): IP_l sends an information $rt_i^{k \prec l}$ to IP_k, where $\mathcal{I}_k$ is the parent of $\mathcal{I}_l$, who can then decide to revoke (in that case we will say that the upwards revocation has been accepted) or not the user, using $rt_i^{k \prec l}$ to retrieve the user's identity for $\mathcal{I}_k$.

Remark 3 (Revocation). *This corresponds to the cascade revocation capability. The goal of the downwards revocation process it to ensure that once a user is revoked of a given domain $\mathcal{I}_l$ then this user is also revoked from all identity domains that are derived from $\mathcal{I}_l$, i.e. the children of $\mathcal{I}_l$ in $\mathbb{I}$, the children of these children, and so on. The optional upwards revocation is there to give the possibility for a domain to signal to the parent domain that a user has been revoked. If this is not executed, IP_k does not learn anything on the identity of the user revoked by IP_l.*

3.2 Security

The main security property that we require from our scheme is that an authentication in a given domain remains anonymous even for the providers of the other identity domains, for instance of the sibling domains in $\mathbb{I}$. We insist on the fact that, in case of revocation, if IP_l does not inform the provider IP_k of the parent identity $\mathcal{I}_k$ of $\mathcal{I}_l$ that a given user is revoked from $\mathcal{I}_l$, then IP_k is not able to know about the identity of this user. We call this property *Cross-Unlinkability* (CU). CU is an adaptation of *Selfless-Anonymity*. Additionally, we directly adapt the security properties a), c) and d) of VLR group signature to our setting of identity management.

3.3 The Construction

We instantiate our algorithms using the CL and BP group signatures, as follows.

KeyGen. IP_0 runs the $\textbf{KeyGen}_{BP}$ algorithm of the BP group signature to generate the public parameters gpk of the scheme. Then each IP_l, including IP_0, creates a key pair (mpk^l, msk^l) compatible with gpk. The msk^l's are kept secret by the IP's. gpk and all the msk^l's are published. The IP's also agree on a set of period tokens $h_{k \prec l}$, that are used for the *Derivation* from $\mathcal{I}_k$ to $\mathcal{I}_l$. We need, for each internal node $\mathcal{I}_k$ in the tree $\mathbb{I}$, to set one period "$k \prec l$" per child $\mathcal{I}_l$ of $\mathcal{I}_k$.

Enrolment. We assume that M_i has fulfilled all the conditions to acquire an identity from the domain $\mathcal{I}_l$. The enrolment phase is then the same as in the BCPZ scheme. M_i is acquired a biometric trait b_i^l. This trait is hashed to form a first part $f_i^l = Hash(b_i^l)$ of his new secret key. M_i and IP_l then jointly run the $\textbf{Join}_{CL}$ algorithm. If $\mathcal{I}_l \neq \mathcal{I}_0$, x_i^l is not chosen randomly by the IP, as in the $\textbf{Join}_{CL}$ algorithm, but it uses the output of the *Derivation* algorithm as the choice for x_i^l, to enable the revocation process. At the end of this algorithm, M_i stores x_i^l, A_i^l and his biometric reference b_i^l. IP_l knows x_i^l and A_i^l and derives the revocation token for M_i for domain $\mathcal{I}_l$: $rt_i^l = x_i^l$.

Authentication. The authentication for a member M_i to a service provider P requiring to belong to $\mathcal{I}_l$ is merely a BCPZ authentication using the group signature parameters for the domain $\mathcal{I}_l$. Concretely, when a user wants to prove to P that he owns an identity, he selects his associated device, connects it to a trusted sensor that communicates with P. The sensor checks using biometrics that the legitimate person is using the card, reads the keys on the card and signs a challenge message sent by P.

Derivation. We now explain how to derive identities. Let $\mathcal{I}_k$ be the parent domain of $\mathcal{I}_l$ in $\mathbb{I}$ and let us assume that a user M_i owns an identity for $\mathcal{I}_k$ and wants to acquire an identity for the domain $\mathcal{I}_l$. M_i has to engage a specific authentication process with the identity provider IP_l.

First, the user authenticates to IP_l, viewed as a service provider for $\mathcal{I}_k$ to prove validity of his identity in $\mathcal{I}_k$ However, he uses the BP signature at period $k \prec l$ instead of the CL scheme. M_i also sends the revocation token $rt_i^{k \prec l}$ corresponding to the $k \prec l$ period. IP_l checks the validity of the signature using $\textbf{Verify}_{CL}$ and checks that the token is the good one using $\textbf{Verify}_{BP}$ with a revocation list set as $\{rt_i^{k \prec l}\}$ (which should fail during the *Revocation Check*). If all tests succeed, IP_l computes $x_i^l = Hash(msk^l || rt_i^{k \prec l})$, which is then used as input for the *Enrolment* algorithm. This derivation process is described in Figure 3.

Remark 4 (Explanations on the Derivation Process)**.** *The BU property of the BP scheme prevents from linking revocation tokens of the same user at different time periods. Here, periods do not represent time, but the different children identities of a given identity domain. Thus, the identities of one user for different domains will not be linkable. Furthermore, the copy of* $rt_i^{k \prec l}$ *kept by* IP_l *is used in the revocation process described below. We consequently achieve our property of Cross-Unlinkability while maintaining a cascade revocation process.*

Notice also that this derivation process at the same time takes into account the parent identities and preserves consistency with the original biometric references, since the new acquisitions have to match with the previous ones.

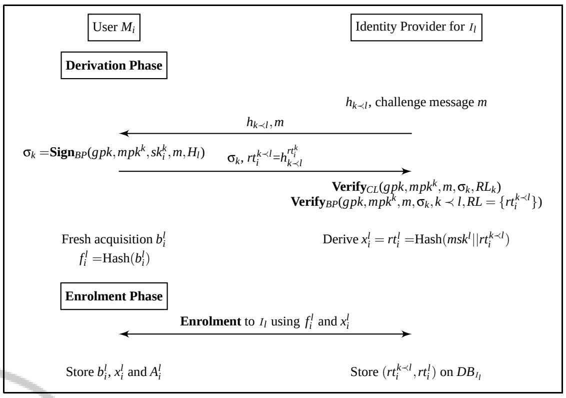

Figure 3: The derivation process.

Revocation. Let us assume that the identity provider IP_l of the identity domain $\mathcal{I}_l$ wants to revoke a user M_i. He proceeds as follows.

Local Revocation: IP_l takes as input the revocation list RL_l and the revocation token rt_i^l of M_i, then adds rt_i^l to RL_l: $RL_l = RL_l \cup \{rt_i^l\}$. The new RL_l is published.

Downwards Revocation: This direction is automatic. All providers for the identity domains $(\mathcal{I}_m)_{m \in M}$ that are children of $\mathcal{I}_l$ learn the revocation token rt_i^l. They all compute $h_{l \prec m}^{rt_i^l}$ and look in their databases $DB_{\mathcal{I}_m}$'s if this token is present. If it is, they start the *Revocation* algorithm for the associated user, using the revocation token rt_i^m associated to $rt_i^{l \prec m}$ in $DB_{\mathcal{I}_m}$.

Upwards Revocation: We recall that this part of the *Revocation* algorithm is optional. IP_l can report to the provider of the parent domain $\mathcal{I}_k$ the user M_i if he thinks that IP_k should revoke him too. He sends to IP_k the item $rt_i^{k \prec l}$ associated to M_i in $DB_{\mathcal{I}_l}$. If IP_k wishes to discover to whom it corresponds, he computes $h_j^{rt_{i'}^k}$ for all the $M_{i'}$'s that belong to $\mathcal{I}_k$. When $h_j^{rt_{i'}^k} = rt_i^{k \prec l}$, the associated user $M_{i'}$ is the user M_i that was revoked by IP_l. IP_k can then, if he desires, revoke $M_{i'}$ from $\mathcal{I}_k$.

ACKNOWLEDGEMENTS

This work is partially funded under the European FP7 FIDELITY project (SEC-2011-284862).

REFERENCES

Boneh, D. and Shacham, H. (2004). Group signatures with verifier-local revocation. In *ACM Conference on Computer and Communications Security*, pages 168–177.

Bringer, J., Chabanne, H., Pointcheval, D., and Zimmer, S. (2008). An application of the Boneh and Shacham group signature scheme to biometric authentication. In *IWSEC*, pages 219–230.

Bringer, J. and Patey, A. (2012). VLR group signatures: How to achieve both backward unlinkability and efficient revocation checks. In *SECRYPT*.

Chen, L. and Li, J. (2010). VLR group signatures with indisputable exculpability and efficient revocation. In *SocialCom/PASSAT*, pages 727–734.