Department of Computer Science,
University of Otago

Technical Report OUCS-2004-17

Further Reflection on Homage Anonymous Group Authentication Protocol

Authors:
Stewart Fleming, Sonil Gohil

Department of Computer Science, University of Otago

Status: Submitted for publication in *Financial Cryptography 2005*

Department of Computer Science,
University of Otago, PO Box 56, Dunedin, Otago, New Zealand

http://www.cs.otago.ac.nz/research/techreports.html

Further Reflection on Homage Anonymous Group Authentication Protocol

Stewart Fleming and Sonil Gohil

Department of Computer Science,
University of Otago,
DUNEDIN, New Zealand.
{stf, sgohil}@cs.otago.ac.nz

Abstract. Anonymous group authentication provides an individual with the ability to prove membership of a group without revealing their identity. The Homage protocol proposed by Handley [9] provides an efficient mechanism for anonymous group authentication. Attacks have been proposed [11] on this protocol which suggest weaknesses in its security. We revisit the original protocol to investigate the nature of the proposed attacks and we propose modifications to the protocol that address them while maintaining the spirit of the original. We then go on to address a remaining weakness in the Homage protocol by considering how non-transferability might be accomplished through the use of biometrics, while preserving anonymity.

1 Introduction

The Homage protocol [9] is a resource-efficient scheme for anonymous authentication of group members. The identity of a group member remains unknown to a certifying group authority when the user is being authenticated. The security of Homage is based on the assumption that the Diffie-Hellman decision problem is hard [3]. The main properties that Homage satisfies are completeness, resource-efficiency, anonymity and a strong disincentive to reveal the private key on which membership is based.

Some questions regarding the security of the protocol were raised by Jaulmes & Poupard [11]. Their concerns are expressed in terms of three attacks on the protocol and their paper concludes with an alternative proposal for fixing the protocol.

Our interest in this area is motivated by the desire to maintain the elegance and simplicity of the original protocol. We investigate the nature of the proposed attacks to determine their feasibility, whether or not they can be avoided *i.e.* if the protocol is fundamentally broken, or if it is simply undergoing iterative refinement [4].

We investigated the nature of the attacks, simulated them using a Java-based implementation of the protocol and determined whether or not the flaws that were uncovered could be avoided. We implemented the original protocol and the proposed attacks. We then implemented the modified protocol that we present here and verified that the attacks were no longer valid. We indicate which attacks can be deflected and where the protocol does need to be strengthened.

The key to anonymity of Homage is that nothing that can reveal the identity of the user or anything known to the authority at registration time is made known at authentication time. However, the authenticity of group membership still rests on the assumption that a group member will not reveal their private key. We believe that disincentives to revealing the private key are insufficient and that the protocol requires extensions to include a mechanism by which the key is linked to biometric information. We include a proposal to extend Homage to include biometric data in such a way as to ensure authenticity whilst preserving anonymity.

2 Related Work

The main focus of our work has been on the proposed attacks on the protocol [11] and we review these attacks before indicating how we believe the original protocol should be strengthened. We also briefly discuss biometric authentication that protects personal privacy through anonymity.

2.1 Wallet Databases Plus Observers

The original work on wallet databases plus observers [5, 6] informs the work on Homage in several different ways. Firstly, the proof of equality of discrete logarithms forms the basis for the zero-knowledge proofs used in Homage. Secondly, the method of preserving anonymity and privacy of personal information is a key influence on Homage and thirdly, we find an interesting direction in the extensions that incorporate biometric information [2,10].

2.2 Privacy Protection Incorporating Biometrics

This latter work is of great interest as it provides both a mechanism to make disclosure of the secret key impossible and support for revocation of credentials, a characteristic which is not directly present in Homage.

The protocol proposed by Impagliazzo & More [10] works as follows. On registration with an authority, biometric data is enrolled from the user and is embedded onto a tamper-proof wallet along with a secret credential k that is chosen by the authority for each discrete time period j. When the user authenticates with the authority, firstly a biometric device on the wallet conducts a biometric authentication with the user to confirm identity before engaging in authentication protocol with the authority. During authentication, the user effectively demonstrates knowledge of the credential k. When issuing new credentials for a new time period, the authority can use a self-healing key distribution method [1, 13]. The protocol provides anonymous group authentication, with a similar level of computation as Homage, but with greater communication requirement.

We address transferability of the user's private key in Homage by including a mechanism for biometric authentication. Non-transferability is more an issue with administration of Homage rather than a fundamental weakness with the protocol itself. The biometric extension to Homage that we include here merely indicates how it is possible to ensure non-transferability with the inclusion of a biometric authentication device in a tamper-proof configuration.

2.3 Homage Group Authentication Protocol

The Homage protocol operates between a group authority and a number of members of that group. The authority issues certificates to group members that they can use to anonymously prove group membership.

Certificates are based on pseudonyms of the public keys of group members, secret values known only to the group authority and random values specific to each group member. The group authority does not need to store any information specific to a group member, nor is any such information useful since the certificate requires knowledge of the private key of the group member before it can be used.

The Homage protocol can be summarized as consisting of three separate activities: authorization, testing the certificate and verification. Testing the soundness of the certificate was proposed so that the group member can be assured that the authority has not cheated in the generation of the certificate.

A prospective group member presents a pseudonym based on his/her public key to the group authority to gain authorization to become a member of the group. The authority chooses a random value *a* and then calculates:

$$\alpha_1 = \left(gy^z\right)^a \left(\mathrm{mod}\, p\right) . \tag{1}$$

$$\alpha_2 = a^w \left(\mathrm{mod}\, p - 1\right) . \tag{2}$$

The pair (α_1, α_2) is the certificate issued to the group member. The certificate is used in anonymous authentication. In order to be authenticated, a group member must provide values based on a valid certificate and demonstrate knowledge of the private key on which the certificate is ultimately based.

To verify that the group authority has not cheated in the issuing of the certificate, the group member should check that it has been computed correctly. The group member goes through the protocol with the group authority as for verification, but the final challenge is omitted. Instead, if the group authority can tell the group member what $g^{cx} = y^c$ is, the group member accepts that the certificate has been correctly computed. Registration requires 4 modular exponentiations.

The proposal in the original protocol for the need for testing the certificate at this point is because the first round of proof is the only point at which the authority can cheat, since zero knowledge is revealed in the second round of proof. This is one of the known ways for an authority to cheat and is eliminated by this verification step – the user can verify the certificate without revealing their identity and if the certificate is not correctly formed, they do not trust the authority.

A user who wants to authenticate with the authority to prove that they are a group member interacts with the authority by choosing $b \in_R \{1..p-1\}, c \in_R Z^*_{p-1}$, then calculating and sending to the authority:

$$\beta_1 = \alpha_1^{cd} \bmod p \ . \tag{3}$$

$$\beta_2 = \alpha_2 v^b (\bmod p-1) \, . \tag{4}$$

$$\beta_3 = y^c (\bmod p) \, . \tag{5}$$

The authority calculates:

$$\gamma_1 = \beta_2^{1/w} (\bmod p-1) \quad . \tag{6}$$

$$\gamma_2 = \beta_1^{1/\gamma_1} (\bmod p) \ . \tag{7}$$

$$\gamma_3 = (\gamma_2 / \beta_3)(\bmod p-1) \, . \tag{8}$$

and sends H(γ3) to the user. The user verifies that $H(\gamma_3) \overset{?}{=} H(y^c \bmod p)$ and then enters a multi-round challenge/response protocol (Table 5) to demonstrate their knowledge of their private key *x*. Once the user has confirmed that values of H(γ3) agree, they are confident that the authority is genuine; once the user has completed *t* rounds of the challenge/response protocol, the authority is confident that the user is authentic, with a less than 1 in 2^t chance of the user cheating. Authentication thus requires *t* + 4 modular exponentiations to complete.

3 Attacks on Homage

The main purpose of our inquiry was firstly to verify the attacks that have been proposed on the protocol and secondly to investigate whether or not the protocol was fundamentally broken. We note firstly that in the review of Homage and proposed attacks [11] there appears to have been some initial confusion between the paper that was presented and the final version that was published [4]. We note their clarifications from the author and have received similar clarifications also [Handley, personal communication 2003].

In fact, one of these clarifications nullifies their basic idea of forging proof of membership through inadvertent revelation of z^th roots modulo *p*. In the verification protocol that is necessary to ensure that the authority has correctly constructed certificates, sending H(γ3) instead of γ3 does not leak any information about *z* that can be used to forge proof of membership. We have considered this additional restriction on the protocol in our work.

3.1 Attack 1: Unsafe Choice of Modulus p

This attack demonstrates that p must be a safe prime. If an authority chooses a modulus p that has more than one prime factor, then they can potentially identify different subgroups of users. A group member can easily detect this form of cheating, by checking that α_2 is relatively prime to $p - 1$, since they know α_2 and p is public.

3.2 Attack 2: Choice of Modulus p with Characteristic Order

If the authority can select $p = 2q + 1$, with $q = 2r_1 r_2 + 1$, where r_1 and r_2 are large prime integers and select parameter $u \in Z^*_{p-1}$ with order r_1, then an attack exists whereby the authority can identify a group member with time complexity linear in the number of registered users. This is the most serious attack proposed on the Homage protocol and is possible since the authority can cache the second half of certificates, and compute the multiplicative order of β_2.

This attack can be detected by noting that it depends on the authority being able to choose $u \in Z^*_{p-1}$ with order r_1. Since u is public, then an authority cannot conceal their intention to cheat. The user can compute the multiplicative order of u in $p - 1$ and hence recover the value of r_2 if u does have order r_1 in $p - 1$; if the authority has cheated in this way, then the multiplicative order of u in $p - 1$ will factor $(p - 1)/2$. If the authority selects u as defined in the restrictions on the protocol, no characteristic multiplicative order can be detected and the attack fails.

3.3 Attack 3: Choosing Secret Key z for Different Subgroups

In this form of attack, an authority can select different secret keys z for different subgroups and hence distinguish users during authentication. This is done by selecting a different secret key z_i for a subgroup and computing certificates as normal. During authentication, the group authority can then check which z_i was used and hence identify the group member. The attack is probabilistic and we were not successful in implementing a simulation that could identify users in a completely reliable fashion.

3.4 Assessment of Attacks

We found that attacks 1 and 2 on the anonymity of authentication were both valid but not significant. Both of these attacks have trivial fixes and one modification was made by the original author in preparing the final draft of the presented paper. Initially, it was *suggested* that $(p - 1)/2$ should have few factors; now, in order to prevent attack 2, it *must* be prime.

We found that attack 3 does pose a significant threat to the security of the protocol. We regard the ability to select different z for subgroups as “cheating by the authority” and address it in our modified protocol below. While Jaulmes & Poupard present a solution to address this attack by modifying the protocol to include bit-commitment schemes, we present an alternative that maintains the spirit of the original protocol and its efficiency.

We chose to defuse attack 3 with a modification to the protocol that forces the authority to prove their knowledge of secret key *z*. We do this as the first stage of authentication and hence assure the user that the key is the same *z* that was used in the generation of the certificate. The protocol for proof of knowledge of *z* provides the authority with no clue as to which z_i to choose, even if he has decided to cheat by associating different z_i with different subgroups. This modification has the advantage of being able to avoid the verification stage.

4 Modified Homage Protocol

The modifications that we have made to the protocol are necessary to address attack 3 above. We have modified the process for issuing certificates at registration and by including a zero-knowledge proof by the authority of secret key *z* as the first stage of authentication. These modifications have the advantage that verification phase can be avoided. In the description that follows, we have included all the restrictions on parameters that are required for the original and modified protocols, in order to avoid the proposed attacks.

- p is a public prime integer such that $q = (p - 1)/2$ is also prime
- g is a public generator of Z^*_{p-1}
- $u \in Z^*_{p-1}$, a public constant co-prime to $p-1$ and able to generate a large fraction, if not all, of Z^*_{p-1},
- A public key of the authority is:

$$v = u^w (\text{mod } p-1) \tag{9}$$

- A public key or pseudonym of the user is:

$$y = g^x \text{ mod } p \tag{10}$$

- H(x) is a publicly-known secure hash function,
- α_1 and α_2 form the certificate issued by the group authority; they require knowledge of private key x to use; α_1 is issued in the form:

$$\alpha_1 = E_1 I_1 (\text{mod } p) \; . \tag{11}$$

$$E_1 = g^a \text{ mod } p \text{ where } a \in_R Z^*_{p-1} \; . \tag{12}$$

$$I_1 = g_1^z (\text{mod } p) \; . \tag{13}$$

$$g_1 = y^a \bmod p\,. \tag{14}$$

$$\alpha_2 = a^w(\bmod p - 1) \text{ as in the original protocol.} \tag{15}$$

- $r_1 \in_R \{1..q\}$, a blinding factor chosen by the user during authentication and:

$$g_2 = g_1^{r_1} \bmod p\,. \tag{16}$$

- $r_2 \in_R \{1..q\}$, a blinding factor chosen by the authority during authentication,
- $r_3 \in_R \{1...q\}$, a blinding factor chosen by the user during authentication,
- z and w are two secret keys held by the authority such that $z \in Z^*_{p-1}$ and $w \in [1, p-2]$. (Jaulmes & Poupard make this restriction based on personal communication with Handley, confirmed by our own exchanges with the author.)
- $x \in Z_{p-1}$ is the private key of the group member.

These parameters are summarized in Table 1 to indicate who knows what and at what point in the protocol they know it.

Table 1. Who knows what (private and shared) in modified Homage and when they know it

	Authority	**User**
Setup	w, z	x
	$p, u, v, y, H(x), g$	
Registration	a	
	$y, \alpha_1, \alpha_2, g_1, I_1, E_1$	
Authentication	$r_2, \gamma_1, \gamma_2, \gamma_3$	$b, c, proof(\mathrm{z}), I_2, r_1$
	$g_2, r_3, x_1, s, H(\gamma_3), \beta_1, \beta_2, \beta_3$	

4.1 Registration

During registration (Table 2), the group member presents their public key (or pseudonym) to the group authority. The variation here on the original is to divide the first half of the certificate α_1 into g_1, E_1 and I_1. This is necessary since, in the authentication phase, we force the authority to prove knowledge of the z that was used to create the certificate. Since z is only used as an exponent, the security of the protocol overall still depends on the difficulty of finding discrete logarithms.

Table 2. Registration of the user with the group authority (modified protocol)

Group Authority		**Group Member**
$g_1 \equiv y^a(\bmod p)$ $E_1 \equiv g^a(\bmod p)$	$\rightarrow g_1, \mathrm{E}_1, \mathrm{I}_1, \alpha_2$	$\alpha_1 = E_1 I_1 \bmod p$

$I_1 \equiv g_1^z (\bmod p)$ $\alpha_2 \equiv a^w (\bmod p - 1)$

4.2 Authentication

We have divided authentication into two parts. The first part constitutes a zero-knowledge proof by the authority of their secret key *z* using the sub-protocol shown in Table 3. The second part is authentication as in the original protocol.

Table 3. Proof of equality of two discrete logarithms

Authority		Member
	$g_2 \leftarrow$	Chooses $r_1 \in_R \{1..q\}$ and calculates: $g_2 = g_1^{r_1} (\bmod p)$ $I_2 = I_1^{r_1} (\bmod p) \equiv \left(g_1^{r_1}\right)^z (\bmod p)$
Choose $r_2 \in_R \{1..q\}$ $x_1 = g_2^{r_2} \bmod p$	$\rightarrow x_1$	
	$r_3 \leftarrow$	Choose $r_3 \in_R \{1..q\}$
$s = r_2 + (r_3 * z) \bmod q$	$\rightarrow s$	$g_2^s (\bmod p) \overset{?}{=} x_1 * I_2^{r_3} (\bmod p)$

There is no way for the authority to be able to cheat by using a different *z* to identify sub-groups of users since they do not get any clue as to which *z* to use at the initiation of the protocol. The authority can work out *a* only if they can compute discrete logarithm from g_2 (16).

The proof of knowledge of the secret key *z* is enabled by issuing α_1 in two parts. Since the part I_1 is based on *z*, if we use this quantity in the zero-knowledge proof, the group member can be assured that not only does the authority know secret key *z*, but that it is the same *z* on which their certificate is based. After the authority has successfully demonstrated proof of knowledge of z, authentication of the certificate proceeds as described in the original protocol (Table 4):

Table 4. Protocol for user to authenticate with group authority in an anonymous fashion

Group Authority		Group Member
$v = u^{w} \pmod{p-1}$ is publicly known.	$\beta_1, \beta_2, \beta_3 \leftarrow$	$b \in_R \{1..p-1\}$, $c \in_R Z^*_{p-1}$. $\beta_1 = \alpha_1^{cd} \bmod p,$ $\beta_2 = \alpha_2 v^{b} \pmod{p-1}$ $\beta_3 = y^{c} \pmod{p}$
$\gamma_1 = \beta_2^{1/w} \pmod{p-1}$ $\gamma_2 = \beta_1^{1/\gamma_1} \pmod{p}$ $\gamma_3 = (\gamma_2 / \beta_3) \pmod{p-1}$	→H(γ3)	$H(\gamma_3) \overset{?}{=} H(y^{c} \bmod p)$

The group member then proves knowledge of private key in γ3 since $\gamma_3 \equiv \beta_3^x$. Using a zero-knowledge proof of equality of discrete logarithm (such as that shown in Table 5), the user can only be authenticated if they can prove knowledge of *x*. Sending H(γ3) is necessary to avoid leaking zth roots modulo *p*, as corrected by Handley in the final draft of the original paper (personal communication, 2003).

Authentication requires *t* + 7 modular exponentiations (as opposed to *t* + 4 for the original protocol), where *t* is the security parameter used in the challenge/response protocol (Table 5). This is the final stage of the protocol used during authentication and consists of *t* rounds of challenges to ensure that the group member has less than 1 in 2^t chance of cheating.

Table 5. Challenge/response protocol for group member to prove knowledge of discrete logarithm γ3in basis β3

Authority		Group Member
	$\gamma_4 \leftarrow$	$t \in_R Z_q^*$ $\gamma_4 = \beta_3^t \bmod p$
$c \in_R \{0,1\}$	$\rightarrow c$	
$c = 0 : \gamma_4 \overset{?}{=} \beta_3^t \bmod p$ $c = 1 : \gamma_3 \overset{?}{=} \gamma_4^{(x/t)} \bmod p$	$c = 0 : t \leftarrow$ $c = 1 : (x/t) \bmod p \leftarrow$	

4.3 Verification

Verification to ensure that the certificate has been correctly computed is not necessary in our modified protocol. This is because the only possibilities for an authority to cheat are in the computation of the certificate and at the first round of proof of knowledge of z during authentication. We examine below why neither of these are possible with the modified protocol.

At authentication, the only information that the authority gets is g_2 (16). If the authority were to cheat, they would have to correctly select the value of z that was used for the user trying to authenticate and prove knowledge of that z. The only way that they can do this is to compute a discrete logarithm.

The authority can no longer cheat in the issuing of certificates using the method described by Handley [9, p306]. If the authority tries to cheat by issuing an incorrectly-formed α_1, then the proof of knowledge of z cannot succeed as the first stage of authentication. The authority cannot select a random value that allows the zero-knowledge proof to complete successfully, since the user expects that I_1 is based on the same value of z that the authority has to prove at this stage. If the authenticating user does not accept the proof of knowledge of z, they will not proceed further with authentication.

An authority *can* still issue a badly-formed second half of the certificate. Say the authority issues (α_1, t) as the certificate where t is random. Now when the user authenticates, they send $(\alpha_1^{cd}, td^w, g^c)$. The authority can recover d and hence extract d^th^ roots of α_1^{cd}. However, since they have not been able to issue an incorrectly-formed α_1, they are not able to recover the identity of the user (again, unless they can compute a discrete logarithm). Hence the user can be assured that certificates are correctly-formed when they are issued and there is no need for the separate stage of verification.

5 Security Review

1. **Protocol** – the modified protocol is now immune to the proposed attacks; attacks 1 and 2 are easily detected by the user so the authority cannot get away with cheating and attack 3 is deflected by our inclusion of the zero-knowledge proof (Table 3) as the first step in authentication and the modified method of issuing certificates. This is a standard zero-knowledge proof and is sound and complete [5, 7, 8]. Proof of knowledge of discrete logarithms (Table 5) is also a standard proof and is zero-knowledge.
2. **Cheating by the authority** – the ability of the authority to cheat by issuing certificates in an incorrect form is removed by issuing α_1 in two parts; selecting unsafe prime modulus is easily detected; selecting different z values for sub-groups is no longer possible due to zero-knowledge proof of z at first stage of authentication.
3. **Cheating by the user** – a user can only forge certificates by computing a discrete logarithm, which we assume is hard, or by computing g^z, which is theoretically possible, but equally hard.

6 Biometric Extensions to Homage

As noted in the introduction, the aspect of Homage that causes us most serious concern is the ability to transfer the private key. While Handley notes that there is strong disincentive based on the fact that all past and future transactions are compromised, this does not make transfer of the private key impossible.

We can easily imagine situations where an authority can grant or deny certificates based on background checks at registration time. If Alice is a user whom the authority will register, but Mallory is not, for Alice to transfer her private key and certificate to Mallory allows Mallory to register when she could not otherwise.

Other systems have been proposed that use biometric identification to address the issue of transferability [2, 10]. However, to directly use biometrics in the implementation would break the anonymity of authentication. We describe here an extension to the implementation of Homage that can be used to assure both anonymity and non-transferability.

The option that we discuss here for biometric authentication and non-transferability of the private key based on the mechanism of wallet databases plus observers [5]. This mechanism uses the invention of the electronic wallet that consists of a computer controlled by the user (a smart-card in this case) and a tamper-proof mechanism embedded within it, known as the observer. The two parts are arranged so that the observer can only communicate with the card and not the outside world (Fig. 1).

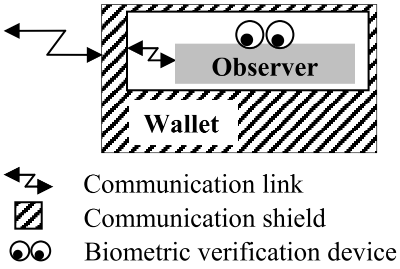

Fig. 1. Wallet databases plus observers with embedded biometric verification [2]

Bleumer's extension to this mechanism [2] is to include a biometric verification facility within the tamper-proof observer. This device is used to verify identity of the controlling user. On registration, biometric credentials are obtained from the user and enrolled on the card along with the user's private key. On authentication, the biometric device compares a biometric sample taken from the authenticating user and compares it with the enrolled user. If there is no match, the device will not engage in further authentication. If there is a match, authentication proceeds as above.

This method has several advantages. Firstly, it provides a mechanism for biometric authentication. Secondly, it protects the identity of the user since no information needs to be sent to the authority at authentication time, other than that the user passed biometric verification. The procedure for obtaining a private key and pseudonym is as outlined in Chaum & Pedersen [5, p11-13].

7 Conclusions

The Homage protocol is an interesting and useful contribution due to its simplicity and efficiency. We have reviewed and implemented the original Homage protocol and assessed the attacks proposed upon it.

Our contributions here are intended to address weaknesses in the administrative aspects of the protocol, to indicate conditions whereby cheating can be detected. We have made some modifications to the Homage protocol as a result and addressed the weaknesses in the protocol that were found. Our modifications are in the spirit of the original proposal and retain its efficiency and anonymity. Through these modifications, we have eliminated the need for verification and removed the opportunities for either the authority or the group member to cheat.

We have also indicated how the administration of the protocol might be improved via a mechanism to ensure non-transferability of the user's private key by incorporation of a biometric authentication device into a tamper-proof electronic wallet.

There is still work to be done to formally prove the security of the protocol and we encourage further iterative refinement as necessary.

References

1. Balfanz, D., Malkin, M., More, S.M.: Sliding-window Self-healing Key Distribution. Presented at ACM Workshop on Survivable and Self-Regenerative Systems. Fairfax, VA. ACM, New York (2003) 60-71.

2. Bleumer, G.: Biometric yet Privacy-Protecting Person Authentication. In: Aucsmith, D. (ed): Proceedings of the Second International Workshop on Information Hiding. Portland, Oregon. LNCS Vol. 1525, Springer-Verlag, Heidelberg (1998) 99-110.

3. Boneh, D.: The Decision Diffie-Hellman Problem. In: Buhler, J. P. (ed): Algorithmic Number Theory: Third International Symposium. Portland, Oregon. LNCS Vol. 1423, Springer-Verlag, Heidelberg (1998) 48-63.

4. Camp, L.J.: Review of Fifth International Conference on Financial Cryptography (FC 2001). In: IEEE Cipher Electronic Newsletter of the Technical Committee on Security and Privacy EI 42 (2001). Online: http://www.ieee-security.org/Cipher/ConfReports/2001/CR2001-FC2001.html, date accessed 17th September 2004.

5. Chaum, D., Pedersen, T.P.: Wallet Databases with Observers. In: Brickell, E. (ed): Advances in Cryptology – CRYPTO '92. Santa Barbara, California. LNCS Vol. 740. Springer-Verlag, Heidelberg (1993) 1-14.

6. Cramer, R.J.F., Pedersen, T.P.: Improved Privacy in Wallets with Observers. In: Helleseth, T. (ed): Advances in Cryptology - EUROCRYPT '93. Lofthus, Norway. LNCS Vol. 765. Springer-Verlag, Heidelberg (1994) 329-343.

7. Feige, U., Fiat, A., Shamir, A.: Zero-knowledge Proofs of Identity. Journal of Cryptology 1 (1988) 77-94.

8. Goldwasser, S., Micali, S., and Rackoff, C.: The Knowledge Complexity of Interactive Proof Systems. SIAM Journal on Computing 18 (1989) 186-207.

9. Handley, B.: Resource-Efficient Anonymous Group Authentication. In: Frankel, Y. (ed): Proceedings of Financial Cryptography 4th International Conference. Anguilla, British West Indies. LNCS Vol. 1962. Springer-Verlag, Heidelberg (2001) 295-312.

10. Impagliazzo, R., More, S.M.: Anonymous Credentials with Biometrically-Enforced Non-Transferability. In: Samarati, P., Syverson P. (eds): Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society. Washington DC, USA. ACM, New York (2003) 60-71.

11. Jaulmes, E., Poupard, G.: On The Security of Homage Group Authentication Protocol. In: Syverson, P. (ed): Financial Cryptography 5th International Conference. Grand Cayman, British West Indies. LNCS Vol. 2339. Springer-Verlag, Heidelberg (2002) 106-116.

12. Staddon, J., Miner, S., Franklin, M., Balfanz, D.: Self-healing Key Distribution With Revokation. In: Proceedings of IEEE Symposium on Security and Privacy. Oakland, CA. IEEE, New York (2002) 241-257.