SCRIPT-ed

Volume3, Issue1, March 2006

Implementing Pseudonymity

Miranda Mowbray[*]

Abstract

I will give an overview of some technologies that enable pseudonymity - allowing individuals to reveal or prove information about themselves to others without revealing their full identity. I will describe some functionalities relating to pseudonymity that can be implemented, and some that cannot. My intention is to present enough of the mathematics that underlies technology for pseudonymity to show that it is indeed possible to implement some functionalities that at first glance may appear impossible. In particular, I will show that several of the intended functions of the UK national ID could be provided in a pseudonymous fashion, allowing greater privacy. I will also outline some technology developed at HP Labs which ensures that users' personal data is released only to software that has been checked to conform to their preferred privacy policies.

DOI: 10.2966/scrip.030106.34

[*] Researcher, HP Labs, Bristol, UK.

1. Introduction

Pseudonymity technology is technology that allows individuals to reveal or prove information about themselves to others, without revealing their full identity. Proving information is, naturally, more difficult than just revealing it, and this paper will concentrate on systems that allow proof of information. I will describe some functionalities that cannot be implemented, and some that can, providing some explanation of the underlying mathematics and protocols without giving all the detail. It is important for policy makers in the area of data privacy to know what is possible and what is impossible, so as to have wide options for policy choice. The results that I will outline in this paper are perhaps not as widely known as they should be.

2. Some things that technology cannot do

Computer technology cannot control data that is in someone's brain. Once you know personal data about me, I cannot use technology to erase that data from your brain, or to prevent you from using it for marketing purposes, or to prevent you from communicating it to others, or to prevent you from combining it with other data that you know and drawing conclusions. Therefore, the standard privacy strategy is to limit how much of my personal data reaches your brain in the first place.

Pseudonymity limits the diffusion of my personal data by releasing different facts about me under different pseudonyms. Only facts about me that are released under the same pseudonym will be easy to combine, and the pseudonymous facts will in general not be linkable to my full identity. Unfortunately, it is not possible in general to determine which set of facts can safely be released under the same pseudonym without revealing the full identity of the data subject. For instance, if an organization comes to know the name and date of birth of the person with a particular pseudonym, this will often be enough to deduce this person's full identity, but it is not possible in general to know whether revealing, say, my date of birth to an organization will enable the organization to deduce my identity. I will ignore this issue for the rest of this paper; when I say that it is not possible to link a pseudonym to a full identity, I will mean that it is impossible to do this in the absence of knowledge of facts about the person with the pseudonym that allow the deduction of that person's identity.

A more technical limitation on the functionality that can be implemented by technology is that any system that revokes privacy on misbehaviour (where misbehaviour is quantitatively defined, that is, it is a matter of human judgement whether misbehaviour has occurred) has to used a trusted organization. In any system which is supposed to revoke my privacy if I misbehave, but only if I misbehave, I have to trust at least that the organization which judges whether I have misbehaved or not will make a correct judgement, and will not falsely condemn me. As I will show later in the paper, some functionalities to do with pseudonymity can be implemented without any trusted organization.

Finally, it is not possible to have both pseudonymity and a full audit trail. A full audit trail links transactions made by the same person to eachother and to that person's full identity. If I always use the same pseudonym when interacting with a particular organization then the organization can link together for audit purposes the

transactions that I make with the organization, but cannot link these transactions to my full identity.

3. Some things that technology can do

The rest of this paper is concerned with functionalities that can be implemented by technology. It may not be obvious at first sight that these are possible, and it is worth knowing that they are, because this opens up more options for policy choice.

First, it is possible to prevent pseudonymous ID-theft without using a shared secret. One way of preventing someone from using your pseudonym - or, indeed, your full identity - and masquerading as you is to use a password or PIN by which you can authenticate yourself. However, when you type in a password or enter a PIN, it is communicated to the machine of the person to whom you are authenticating yourself, and you have to trust that person to keep it secret and not to misuse it. But it is possible for you to authenticate yourself using a private key which unlike a password or PIN *never leaves your computer*. Identity theft is one of the problems that national ID cards are supposed to address. The authentication method using a private key allows you to prevent theft of your identity with greater privacy than you would have if ID cards were used for authentication, because when you authenticate yourself using a private key you do not necessarily reveal your full identity (or an ID number that can be traced back to your full identity) to the person to whom you are authenticating yourself.

Second, it is possible to set up a system of unique unlinkable psuedonyms. In this system, you can only have one pseudonym per organization, but *no-one* can link your pseudonyms to eachother or to your real identity, even if all the organizations in the system conspire against you. This has a possible application for instance in the processing of medical data. Data bases in the Scottish NHS use identifiable data because the Scottish NHS wishes to avoid duplicate records. Medical data is typically stored under a patient number rather than (say) name and date of birth, but is identifiable in the sense that it is possible to trace back from the patient number to the patient's full identity. If unique unlinkable pseudonyms were used instead of patient numbers or name and date of birth, this would still allow data base administrators in the Scottish NHS to eliminate duplicate medical records, because each patient would have a unique pseudonym with the medical organization, but it would ensure that the pseudonym could not be traced back to the patient's full identity.

Third, it is possible to have an anonymous credential. An anonymous credential is a proof about some fact about one of your pseudonyms which does not reveal either this pseudonym or my your identity. For instance, an anonymous credential system could make it possible to prove that you have a driving licence without revealing either your full identity or your driver number. Other possible applications of an anonymous credential system are to allow you to prove that you are not on a national watch list for terrorists or paedophiles, or that you are not an illegal immigrant, without proving your identity. This would allow more privacy to those not on the national watch lists than if proving these facts required full identification using a national ID card.

Fourth, it is possible to ensure that your personal data is released only to software that has been checked to conform to particular privacy policies - where you decide which privacy policy you like, and where the policy that you specify may be different for different sets of your personal data.

I will now outline the mathematics that makes these functionalities possible. Several of the basic protocols are presented in a twenty-year-old paper by David Chaum. Although there have been considerable developments since then, it is perhaps surprising that these technical possibilities are not more widely known by now. One possible reason is that some of these possibilities are counterintuitive, and hence hard to believe or remember unless you have seen the underlying mathematics.

3.1 Preventing identity theft without a shared secret

The fundamental mathematical tool that everything that I describe in this paper built on is *public key cryptography*.[1] In traditional cryptography, the security of a message relies on a shared secret; the sender and receiver of the message both know a secret key, which they use both to encrypt and to decrypt messages that they send eachother. In public key cryptography, in contrast, the key used for encryption is not the same as the key used for decryption. The key used for encryption of messages addressed to a person U can be made public, while U keeps the key for decrypting these messages private. Anyone who knows the public key can send a message to U securely, without having a shared secret with U – in particular, without knowing U's private key.

Public key cryptography relies on the existence of one-way encryption functions, that is, mathematical functions f_U which can be used to encrypt a message m (written as a binary number) into an encrypted message $f_U(m)$, such that it is easy to compute $f_U(m)$ if you know m and the public key of U, but is hard to compute m given $f_U(m)$ unless you know U's private key. One set of one-way encryption functions is the set of functions $f_U(m)=m^n \pmod r$, where $n = p.q$, p and q are good primes, U's public key is n and U's private key is the pair of numbers (p,q). There are other one-way encryption functions that can be used if in the future someone discovers an easy way of computing m given $f_U(m)$ for this family of encryption functions. A property of functions in this family (and of some other one-way encryption functions) that I will use in this paper is that they are *multiplicative*, that is, if you multiply two messages together and encrypt the result, what you get is equal to the number you get if you encrypt the two messages and multiply the encrypted messages together. In mathematical notation, $f_U(m).f_U(m')=f_U(m.m')$. (Strictly speaking, it is equal up to a multiple of r. All the arithmetic in this paper is modulo r.)

Since only U knows U's private key, and it is hard to decrypt a message without knowing the private key, this gives a way for U to digitally sign a document. U's signature on a document c is $f_U^{-1}(c)$, the result of decrypting c using U's private key. Moreover, it gives a way for U to authenticate himself. The basic idea is that U receives a challenge number c from the person or organization to which he wishes to authenticate himself, and replies by sending back $f_U^{-1}(c)$. The challenger can check using U's public key that this number encrypts to c, so that this is indeed U's signature on c, and hence could only have been computed easily by someone in possession of U's private key. U has now proved his identity to the challenger (or, at least, U has proved to the challenger that he possesses the private key) although there is no secret shared between U and the challenger.

---

[1] W Diffie and M Hellman, "New directions in cryptograpy", (1976) 22 *IEEE Transactions on Information Theory* 644

Actually, in order to prevent various types of attacks, the full protocol is more complicated than that: U appends a one-time random number and timestamp to the challenge, signs the result, appends the random number and timestamp, encrypts the lot in the challenger's public key, and sends the result to the challenger, and there is a certificate authority who signs U's public key to prevent a man-in-the-middle attack. However, the basic idea is just that U signs the challenge. I mention the complications here just to emphasize that in this paper I will be just giving the central ideas that make the protocols work, rather than all the details.

One way of setting up a pseudonym system is for U to select a one-way encryption function with which to communicate with a particular organization, and to use the public key for that encryption function as his pseudonym with that organization. The pseudonym is "public" in the sense that it is known both to U and to the organization, rather than being private to U, but in general it might not be known by anyone other than U and the organization. U and the organization can use authentication with U's private key to prevent an identity thief from pretending to the organization to be the person with U's pseudonym. U can use other encryption functions – and hence other pseudonyms – to communicate with other organizations.

3.2 Unique unlinkable pseudonyms

With public key cryptography, it is possible to have *blind signatures*, that is, an organization can sign a number without knowing what that number is. Here is how it works.[2]

Suppose U would like to have authority A's signature on a number. U picks a number n of a special form such that it is hard for anyone but A to find a number m such that $f_A(m)$ is of that form – for instance, the form might be a long digital number with the first half of the number identical to the second half. U also picks a number b, which can be any number for which U knows the modular inverse b^{-1}; this will be used to "blind" the number n. U calculates $f_A(b).n$ and sends this to A, asking A for a signature. If A agrees, A sends back $f_A^{-1}(f_A(b).n)$, which by the multiplicative property is equal to $b.f_A^{-1}(n)$. U multiplies this by b^{-1} to get $f_A^{-1}(n)$, which is the signature of A on n. Anyone who knows A's public key can check that $f_A^{-1}(n)$ encrypts to a number n of the special form, so that U must have obtained it through A; but the protocol does not reveal n to A.

This blind signature protocol was invented by David Chaum, and he used it produce a system of unlinkable pseudonyms.[3] Here is a sketch of the basic protocol. There is an authority A which knows U's real identity, but which will not know any of U's pseudonyms. U would like to communicate pseudonymously with an organization B. U asks A for a one-time registration key for B, and using the blind signature protocol, B obtains the registration key $f_A^{-1}(n)$ from A where n is of the special form. Later, U generates a new public key/private key pair to use with B, and sends $f_A^{-1}(n)$ and the public key n_B to B. B checks that n is of the special form. If it is, B checks with A that

---

[2] D Chaum, "Security without identification: transaction systems to make Big Brother obsolete", (1985) 28(10) *Communications of the ACM* 1030 (hereafter referred to as Chaum, "Security without identification").

[3] Chaum, "Security without identification".

n has not previously been used to register. This ensures that U can only have one pseudonym with B.

U can use this protocol to register pseudonymously with many organizations, using a different pseudonym with each organization. The organizations and A cannot link any of the different pseudonyms to eachother or to U's real identity, even if they all conspire. Optionally, the system can be set up so that the pseudonyms periodically time out – for instance, U's pseudonyms might time out once a year, and U would then get a brand new set of pseudonyms, all unlinkable to the old pseudonyms. This allows for a clean technological solution to some knotty practical problems in implementing legal requirements that limit the period of retention of personal data.

If U uses a different pseudonym and a different private key for each organization in the system that he interacts with, this means that U will have to recall all these numbers. This is not as arduous as it might seem, because there is technology which can help. For instance, single sign-on software enables U to have single sign-on password for his own computer which then manages all his pseudonyms and keys for him, and special hardware can be used for secure storage of private keys. The security and privacy of the system for U relies on the secrecy of U's private keys. However, at least the protocol does not require these keys to communicated to anyone else. So U can concentrate on keeping the keys that are in his own possession safe, without having to worry (as he would have to worry in a system using passwords or PINs for authentication) about copies of this information that are in the possession of someone else.

In the pseudonym system described in this section the pseudonyms are unique, that is, U cannot have more than one pseudonym with the same organization. The uniqueness property is necessary for some applications – for instance, avoiding duplicate records in medical databases, or preventing benefit fraudsters from claiming the same benefit multiple times, or linking together for audit purposes all transactions with the organization made by the same person, or producing the "good-behaviour" credentials that I will describe later. In situations where it is not necessary for the system to enforce uniqueness of pseudonyms, then U does not need use an authority A who knows his identity in order to create pseudonyms. U can simply pick a public key/private key pair to use with B, and use the public key as his pseudonym with B. The organization B no longer can be sure that two different pseudonyms correspond to two different people – indeed U can use multiple pseudonyms with B, each with its own corresponding private key, if he wishes. But it is still the case that no-one (other than U) can link any of U's pseudonyms to eachother or to U's real identity. For the subsequent sections of this paper I will assume that pseudonyms are unique.

3.3 Anonymous credentials

It may happen that U uses different pseudonyms with organizations B and C, but would like to prove to B some fact about his interactions with C. An example suggested by Jan Camenisch and Anna Lysyanskaya[4] is pseudonymous car rental.

---

[4] J Camenisch and A Lysyanskaya, "Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation", in B. Pfitzman (ed), *Advances in Cryptology—EUROCRYPT 2001*, 2045 *LNCS* (2001), 93 (hereafter referred to as Camenisch and Lysyanskaya, "Non-transferable credential system").

Suppose that U uses pseudonyms n_B, n_C with a car rental agency B and a driving licence agency C, and would like to prove to the car rental agency that he has a driving licence without allowing the car rental agency to link the two pseudonyms.

One method of doing this is to use a trusted pseudonym-linking organization P. U shows the driving licence for n_C to P, and also proves by answering challenges from P that he knows the private keys corresponding to both n_B and n_C . P then vouches that n_B has a driving licence. The drawback of this method from the point of view of privacy is that U has to trust P not to abuse the ability to link the pseudonyms n_B and n_C. Note that P cannot link these to any other pseudonyms of U, or to the real identity of U, or to any future car hire or driving licence pseudonyms that U obtains after n_B and n_C time out. Nevertheless there is still potential for U's privacy to be diminished if P is not trustworthy.

There is, however, a method by which U can prove he has a driving licence without letting *anyone* link his pseudonyms. The protocol that I will now outline for doing this is once again by David Chaum.[5] Several researchers have subsequently developed other protocols that improve on this one, but I have chosen to outline Chaum's original protocol for anonymous credentials because it is relatively simple to understand.

3.3.1 Anonymous credential system

As before, A is an authority that knows U's real identity but none of U's pseudonyms with other organizations. In the anonymous credential system, the credential for U issued by C is $f_C^{-1}(n_A)$, the signature of C on the public key that U uses with A. For instance, if C is a driving licence authority, this credential acts as a certificate from C that U has a driving licence. U obtains this credential from C using a blind signature protocol, so that C does not learn n_A. In order to obtain it, U first authenticates himself to C using the private key associated with n_C, so that C can check that he really does have a driving licence before issuing the credential.

It is not a good idea for U to show the credential directly to B, because anyone who sees this credential and knows C's public key can encrypt the credential using C's public key to reveal n_A, which can be linked by A to U's real identity. Therefore, U will prove to B that he has the credential without displaying it directly. To prove this, U will show to B a number $f_C^{-1}(n)$ for some n, together with a certificate from A that n is of the form

$$n = n_A.f_C(s) \text{ for some } s=hash(t),\ t \text{ known to U}$$

where *hash* is a mathematical function that has the property that it is easy to calculate $hash(t)$ given t, but given s it is very difficult to find t such that $hash(t)=s$. Clearly, n is related to n_A, and the proof involves showing a signature by C on n instead of on n_A. It can be shown that U can only produce this proof if U does indeed have the credential.

U obtains the proof as follows. For several different values of t, chosen at random by U, U computes the corresponding value of $n = n_A.f_C(hash(t))$. U sends these numbers n in blinded form to A, using a different blinding number for each n. A selects a subset of the numbers that U has sent, and challenges U to demonstrate that these are

[5] Chaum, "Security without identification".

of the correct form by revealing the blinding number and *t*. If U can do this for enough of the numbers, A accepts that one that has not been selected is also of the correct form, and certifies it, still blinded. U removes the blind to obtain the certificate from A for *n*, where $n = n_A.f_C(s)$ for some *s*. Because the certification was done in a blinded form, authority A has learned neither *n* nor *s*. U now multiplies the credential $f_C^{-1}(n_A)$ by *s* to obtain $f_C^{-1}(n_A).s$, which by the multiplicative property for f_C is equal to $f_C^{-1}(n_A.f_C(s)) = f_C^{-1}(n)$. U now has both parts of the proof that he has the credential.

In this explanation I have assumed that C awards just one type of credential. In general, C might award many different types of credential, each signed using a different private key. So for instance a signature of n_A that is signed using one particular key of C might certify that U has a heavy goods vehicle licence expiring in October 2010, and a signature by C using a different key would signify a different kind of driving licence.

3.3.2 Pseudonymous accountability

If U misbehaves using his pseudonym with B, other organizations might want to know. A certain amount of pseudonymous accountability can be achieved by requiring good-behaviour credentials. For instance, U might obtain an anonymous credential from a car hire agency certifying that he did not crash any hired cars last year. This might be interesting to other organizations, or to the agency itself when U's current pseudonym times out. Since U can only have one pseudonym with the car hire agency at a time, it is not possible for U to crash a car under one pseudonym and get a good-behaviour credential covering the time period of the crash under another pseudonym. Good-behaviour credentials could also be used for example to allow U to prove that he was not on a national watch list for paedophiles or terrorists, without revealing his full identity. The administrators of the watch list would know the full identities of the people on the list, but if U was not on the list the administrators could grant him an anonymous credential certifying this.

Camenisch and Lysyanskaya point out, however, that good-behaviour credentials do not provide enough accountability for some types of application, because for some applications it is not enough to check that U has behaved well in the past, it is also desirable to be able to remove or reduce U's privacy if he behaves badly in the future. This can be done with the use of trusted pseudonym-linking organizations. In their car hire example, the car hire organization would probably not allow U to hire a car without first authenticating his car hire and driving licence pseudonyms to a pseudonym-linking organization P, and authenticating his car hire pseudonym and revealing his real identity to a pseudonym-linking organization Q, which might be a separate organization from P. If U breaks the speed limit in his hired car, then P can link his car hire number to his licence number, and contact the driving licence agency so that a point is put on his licence. If U runs over someone in his hired car, then Q can link his car hire number to his real identity, so that he can be made answerable for this under the legal system.[6] Of course, U has to trust that P and Q will not abuse their pseudonym-linking powers. However, as I mentioned in earlier in this paper, if a system allows revocation of U's privacy on misbehaviour, U has to trust some organization in the system.

---

[6] J Camenisch and A Lysyanskaya, "Non-transferable credential system".

It might be argued that the car rental example is unrealistic, because in practice no car rental agency is going agree to rent a car to someone whose full identity they do not know. However, this technology can be used for application in areas where the organizations involved are more likely to agree to pseudonymity.

3.3.3 Preventing transfer of credentials

Chaum's anonymous credential system has an embarrassing property that appears not to have been noticed until several years after the protocol was published: his credentials are transferable. If U wishes, he can give (or sell) to someone else the credential that he has a driving licence, together with the private key that he uses to authenticate himself to the driving licence authority. Anyone in possession of these will be able to masquerade as U to the driving licence authority and "prove" that she has a driving licence, even if she has never passed a driving test. Fortunately, there are several solutions to this problem. I will describe three approaches.

A first approach is to design an anonymous credential system in which credentials are not transferable. Lysyanskaya, Rivest, Sahai and Wolf[7] have done some work on this; they have designed a practical system for non-transferable anonymous credentials, which however has the limitation that credentials can only be used once. This system would not be appropriate for driving licences, but is appropriate for some other applications.

A second approach, suggested by Bleumer,[8] is to incorporate the hash of a biometric in the pseudonym. When U claims to be the person who uses a particular pseudonym with the driving licence authority, this can be checked using a device with tamper-resistant hardware that reads U's fingerprint (or other biometric), calculates the hash, and compares it with the pseudonym. The hash function has the properties that from the hash of the fingerprint it is not possible to deduce the fingerprint, and that it is unlikely that U will be able to find someone else whose fingerprint hashes to the same as his. Note that the biometric is just used to make it difficult to transfer the credential, it is *not* used as a unique personal identifier. Therefore this approach does not require the biometric to be unique. Indeed, in several people may have the same hashed fingerprint as U. However, since the number of people in the world with the same hashed fingerprint as U will be small, it is very unlikely that U will be able to find one such person who is also interested in buying his driving licence. The unique personal identifier for U that is used by the driving licence authority is U's pseudonym, which is derived from both the hashed biometric and the public key that U uses to communicate with the driving licence authority, and the protocol for generating pseudonyms ensures that the pseudonym is indeed unique. U authenticates himself as the person with the pseudonym using his private key, with the biometric check as an additional check to prevent U from transferring his credentials or private keys. The hash can be calculated using a different hash key for each organization, so that the incorporation of the hashed biometric in the pseudonyms does not allow different pseudonyms for U to be linked.

---

[7] A Lysyanskaya, R Rivest, A Sahai and S Wolf, "Pseudonym systems", in *Selected Areas in Cryptography*, 1163 *LNCS* (1999), 302.

[8] G Bleumer, "Biometric yet privacy protecting person authentication", in D Aucsmith (ed), *Information Hiding 1998*, 1525 *LNCS* (1998), 99.

Camenisch and Lysyanskaya[9] have an interesting third approach to preventing transfer of credentials. Their system uses a technique called circular encryption to ensure that if U transfers any one of his credentials or pseudonyms to another person, then that person can use all of U's credentials and pseudonyms, and can take over U's entire system identity. If for instance U's bank account is part of the system, this may be sufficient deterrent to prevent U from selling someone else the ability to use his driving licence.

3.4 Release personal data only to specified software

Researchers at HP's Trusted Systems Lab have been developing technology that can allow U to ensure that his personal data is only processed by software that has been checked to conform to a privacy policy, where U specifies which privacy policy he requires.[10] This functionality is implemented using trusted computing platforms, for which the Trusted Computing Group[11] (an industry standards organization) has been developing standards. U's computer releases U's data in the form

$$f(g(\text{pseudonymous data}))$$

where f and g are encryption functions. Only computers with a credential that they are one of a set of trusted computing platforms can decrypt f, and on trusted computing platforms only a specific set of software programmes, chosen by U, can decrypt g: in particular, U may require that only programmes which have been checked to conform to a certain privacy policy can decrypt g. U can choose different privacy policies for different sets of his data. The trusted computing platforms check the state of their software on start-up and before releasing user data, to check that there is no Trojan in the system. User data is re-encrypted before being released onward to other computers.

The credential that a machine is a trusted computing platform cannot be transferred to another machine, because it is derived from a secret ID encoded in tamper-resistant hardware built into the machine, using an anonymous credential protocol invented by Brickell, Camenisch and Chen.[12] Credentials of rogue machines can be revoked. There will be multiple credential-granting organizations, so that the system does not rely on a single credential-granting organization.

Despite this infrastructure, there are still aspects of the system that U has to trust. U has to trust that the environment is secure, for instance that no-one is standing behind him reading the screen and taking notes as he types his personal data into his computer. U has to trust that the credential-granting organization that he chooses is not corrupt or incompetent, and that the people who check that the software conforms to his chosen privacy policy perform their checks accurately. (If U is very suspicious, he could permit his data to be released only to software that he has checked himself, but in that case he would have to trust his own competence in checking the software.) U has to trust that the infrastructure has been correctly implemented. Finally, if U's

---

[9] J Camenisch and A Lysyanskaya, "Non-transferable credential system".

[10] M Cassasa Mont, S Pearson and P Bramhall, "Towards accountable management of privacy and identity information", in Dieter Gollman and Einar Stekkenes (eds), *Computer Security – ESORICS 2003*, 2808 *LNCS* (2003), 146.

[11] Trusted Computing web site (*https://www.trustedcomputinggroup.org/home*)

[12] E Brickell, J Camenisch and L Chen, "Direct anonymous attestation", *Proceedings of the 11th ACM Conference on Computer and Communications Security* (2004) 132.

data reaches a human being after its processing by software, U has to trust that this person will treat the data properly, since computer technology cannot control data in a human brain. These aspects of the system cannot be enforced through technological means, so there is still a role for the lawyers.

4. Conclusion

In this paper I have shown that several functionalities that are currently implemented in a way that requires proof of identity could be implemented instead in a pseudonymous fashion, enabling a greater degree of privacy. In particular, I have shown that this is the case for several of the issues that the UK national ID card is intended to address. This does not mean that all functions of the national ID card could or should be implemented pseudonymously. The use of different unlinkable pseudonyms with different government organizations would hamper correlation of data between these organizations (although some correlation could be done with the use of credentials) and hence potentially weaken the government's ability to provide high quality "joined-up" service to citizens. It does, however, mean that some of the arguments that have been put forward for the necessity of non-pseudonymous UK ID cards are unconvincing from a technical point of view.

As a final remark, this paper describes several data privacy issues that cannot be addressed by technology, as well as several that can. Although technology can provide a surprising amount of privacy-related functionality, technology by itself cannot ensure data privacy. It is necessary to have appropriate laws and norms to deal with the issues that cannot be addressed technologically, and to provide backup for when technological approaches fail.

Acknowledgment

Thanks to Graeme Proudler of the Trusted Computing Group, to Lilian Edwards and Nadine Eriksson-Smith who organized the AHRC Research Centre's 2005 workshop on Privacy and Technology, and to the workshop attendees whose comments contributed to this paper.