Saving On-Line Privacy

Jan Camenisch, Gregory Neven

Saving On-line Privacy

Jan Camenisch and Gregory Neven

IBM Research – Zurich
{jca,nev}@zurich.ibm.com

Abstract. With the increasing use of electronic media for our daily transactions, we widely distribute our personal information. Once released, controlling the dispersal of these information is virtually impossible. Privacy-enhancing technologies can help to minimize the amount of information that needs to be revealed in transactions, on the one hand, and to limit the dispersal, on the other hand. Unfortunately, these technologies are hardly used today. In this paper we aim to foster the adoption by providing a summary of what such technologies can achieve. We hope that by this, policy makers, system architects, and security practitioners will be able to employ privacy-enhancing technologies.

1 Introduction

The number of professional and personal interactions we are conducting by electronic means is raising daily. These on-line transactions range from reading articles, searching information, buying music, and booking trips, to peer-to-peer interactions on social networks. Thereby we reveal a plethora of personal information not only to our direct communication partners but also to many other parties of which we are often not even aware. At the same time, electronic identification and authentication devices are becoming more and more widespread. They range from electronic tickets, toll systems, to eID cards and often get used across different applications.

It has become virtually impossible to control where data about us are stored and how they are used. This is aggravated as storage becomes ever cheaper and the fact that the increasingly sophisticated data mining technologies allow for all of these data to be used in many ways that we can not even imagine today.

It is thus of paramount importance to enable individuals to protect their electronic privacy. Luckily, there exists a wide range of privacy enhancing technologies available that can be used to this end. These range from privacy-aware access control and policy languages to anonymous communication protocols and anonymous credential systems. The PRIME (Privacy-Enhancing Identity Management for Europe) project [pria] has shown that these technologies can indeed be used together to build a trust and identity management systems that allows for protecting one's on-line privacy and that they are ready to be applied in practice. The PrimeLife project [prib] has taken these research results up and is concerned with bridging the gap from research to practice.

Let us, however, note that while technology can help, the users also need to learn about the perils our digital world and how to guard their privacy. Of course, ICT systems must to this end provide sufficient information to the users about what is happening with their data.

It seems that making use of privacy-enhancing technologies is harder than for other security technologies. One reason for this might be that the properties that they achieve are often counter intuitive, in particular in case of cryptographic building blocks. In an attempt to foster the adoption of privacy-enhancing technologies (PETs), we overview in this paper the most important cryptographic PETs and summarize what they achieve. We also give references for technical details of them. Finally, we explain how these technologies can be embedding into larger systems.

2 Cryptography to the Aid

There is a large body of research on specific cryptographic mechanisms that can be used to protect one's privacy. Some of them are theoretical constructs, but many are actually completely practical and can be readily applied in practice. We here concentrate on the latter ones.

The oldest privacy-protecting cryptography are of course encryption schemes by themselves: they allow one to protect information from access by third parties when data is stored or when sent to a communication partner. There are, however, a number of variants or extension of such basic encryption that have surprising properties that can offer better protection in many use cases as we shall see. Apart from encrypting, one often needs to authenticate information. Typically, this is done by using a cryptographic signature scheme. The traditional signature schemes typically provide too much authentication in the sense that they are used in a ways that reveals a lot of unnecessary contextual information. The cure here is offered by so-called anonymous credential schemes and their extensions which we will present. Finally, we briefly discuss a number of cryptographic applications such as electronic voting schemes and privacy-enhanced access control schemes.

2.1 Private Credentials, Their Extensions, and Applications

Certified credentials form the cornerstones of trust in our modern society. Citizens identify themselves at the voting booth with national identity cards, motorists demonstrate their right to drive cars with driver licenses, customers pay for their groceries with credit cards, airline passengers board planes with their passports and boarding passes, and sport enthusiasts make their way into the gym using their membership cards. Often such credentials are used in contexts beyond what was originally intended: for example, identity cards are also used to prove eligibility for certain social benefits, or to demonstrate to be of legal age when entering a bar.

Each of these credentials contains attributes that describe the owner of the credential (e.g., name and date of birth), the rights granted to the owner (e.g., vehicle class, flight and seat number), or the credential itself (e.g., expiration date). The information in the credentials is trusted because it is certified by an issuer (e.g., the government, a bank) who on its turn is trusted.

There is a number of different ways how such credentials can be technically realized. Depending on their realization, they offer more or less protection of the user's privacy. For instance, they are often realized by an LDAP directory maintained by the issuer. That means that the user wants to use a credential with some party (the verifier), the user will have to authenticate, typically with a username and password towards the verifier who will then look up the user's credentials in the LDAP directory. While this realization might satisfy the security requirement of the verifier and the issuer, it offers virtually not protection to the users. Apart from username/password being a rather insecure authentication mechanism, the user has 1) no control which information the verifier request from the issuer and 2) the issuer learns with which verifier the user is communicating.

A better realization of credentials is with certificate with so-called attribute extensions [CSF+08]. Here, the user chooses a public/secret key pair and then obtains a certificate from the issuer on her public key. The certificate includes all statements that the issuer vouches for about the user. The user can then send this certificate to the verifier together with a cryptographic proof of ownership of the secret key. The user knows which data is revealed to the verifier by the certificate, but has to reveal all of the included attributes so that the verifier can check the issuer's signature. Moreover, if the verifier and the issuer compare their records, they can link the user's visit to the issuing of the credential by simply comparing the issuer's signature.

Anonymous credentials [Cha81,Bra99,CL01] (often also called private credentials or minimal disclosure tokens) solve all these problems and indeed offer the best privacy protection possible while offering the same cryptographic security. They work quite similarly to attribute certificate, the difference being that they allow the user to "transform" the certificate into a new one containing only a subset of the attributes of the original certificate. This feature is often called *selective disclosure*. The issuer's signature is also transformed in such a way that the signature in the new certificate cannot be linked to the original signature; this is usually called *unlinkability* in the literature.

Extended Functionalities. Apart from the basic features of selective disclosure and unlinkability sketched above, many anonymous credential systems offer additional features that can be very useful in practical use cases. In the following, we discuss the most important of these features.

Attribute Properties. Rather than revealing the complete value of an attribute, some credential systems allow the user in the transformation to apply any (mathematical) function to the original attribute value. For instance, if the original

certificate contains a birthdate, the transformed attribute could contain only the decade in which the user was born. As a special case, the function could be boolean (meaning, having as output "true" or "false"), so that only the truth of a statement about the attribute is revealed. For instance, based on the birthdate in a certificate, the user could prove that she is between 12 and 14 years old. The schemes also allow for logical AND and OR combinations of such boolean expressions [CDS94].

Verifiable Encryption. This feature allows one to prove that a ciphertext encrypts a value that is contained in a credential. For instance, a service provider could offer its service to anonymous users provided that they encrypt their name and address as contained in their identity card under the public key of a trusted third party, such as a judge. The cryptography ensures that the service provider cannot decrypt the name and address himself, but can rest assured that the ciphertext contains the correct value. In case of misuse of the service, the service provider or a law enforcement agency can then request the third party to decrypt the user's name and address from the ciphertext. Note that it can be decided at the time of showing the credential whether or not any information in the credential should be verifiably encrypted, i.e., this need not be fixed at the time the credential is issued and can be different each time a credential is shown.

An essential feature that we require in this setting from an encryption scheme is that of a label [CS03]. A label is a public string that one can attach to a ciphertext such that without the correct label, the ciphertext cannot be decrypted. The most common usage for the label in our setting is to bind the conditions and context under which the trusted third party is supposed to decrypt (or not decrypt) a given ciphertext.

In principle, one can use any public encryption scheme for verifiable encryption [CD00]; the most efficient way to do so however is probably using the Paillier encryption scheme [Pai99] for which efficient proof protocols and an variant secure against chosen-ciphertext attacks exist [CS03]. Security against chosen-ciphertext attacks is actually crucial in this setting: the thrusted third party's jobs is essentially a decryption oracle and hence semantic security would not be sufficient.

Revocation of Credentials. There can be many reasons to revoke a credential. For example, the credential and the related secret keys may have been compromised, or the user may have lost her right to carry a credential. Also, sometimes a credential might only need to be partially revoked. For instance, an expired European passport can still be used to travel with Europe, or a driver's license revoked because of speeding could still be valid to prove the user's age or address.

Possible solutions to revocation in the case of non-anonymous credentials is to "blacklist" all serial numbers of revoked credentials in a so-called *certificate revocation list* [CSF+08] that can be queried on- or off-line, or to limit the lifetime of issued credentials by means of an expiration date and periodically re-issue non-revoked credentials. The latter solution works for anonymous credential as well,

even though re-issuing may be more expensive than for ordinary credentials. The former solution as such does not work, as revealing a unique serial number of a credential would destroy the unlinkability property. However, the general principle of publishing a list of all valid (or invalid) serial numbers can still work if, rather than revealing their serial number, users leverage the attribute property feature to prove that it is among the list of valid serial numbers, that it is not among the invalid ones. A number of protocols that are work along these lines have been proposed [BS04,BDD07,NFHF09] where the solution by Nakansihi et al. [NFHF09] seems to be the most elegant one.

Another solution inspired by revocation lists is the use of so-called dynamic accumulators [CL02,CKS09]. Here, all valid serial numbers are accumulated (i.e., compressed) into a single value that is then published. In addition, dynamic accumulators provide a mechanism that allows the user to prove that the serial number of her credential is contained in the accumulated value. Whenever a credential is revoked, a new accumulator value is published that no longer contains the revoked serial number. The schemes require, however, that users keeps track of the changes to the accumulator to be able to execute their validity proofs.

We observe that enabling revocation brings along the risk that the authority in control of the revocation list (or accumulator value) modifies the list to trace transactions of honest users. For instance, the authority could fraudulently include the serial number of an honest user in the revocation list and then check whether the suspected user succeeds in proving that her credential is not on the list. Such a behavior could of course be noted by, e.g., a consumer organization monitoring changes to the public revocation values.

One idea to lessen the trust that one has to put into such a third party is by using threshold cryptography, i.e., by distributing the power to update the revocation list over multiple entities such that a majority of them is needed to perform an update.

Limited-use credentials. Some credentials, such as entrance tickets, coupons, or cash money, can only be used a limited number of times. A very basic example of such credentials in the digital world is anonymous e-cash, but there are many other scenarios. For instance, in an anonymous opinion poll one might have to (anonymously) prove ownership of an identity credential, but each credential can only be used once for each poll. Another example might be an anonymous subscription for an on-line game, where one might want to prevent that the subscription credential is used more than once simultaneously, so that if you want to play the game with your friends, each friend has to get their own subscription [CHK$^+$06].

When implementing a mechanism to control the number of times that the same credential can be used, it is important that one can define the scope of the usage restriction. For instance, in the opinion poll example, the scope is the specific poll that the user is participating in, so that participating in one poll does not affect his ability to participate in another one. For electronic cash, on the other hand, the scope is global, so that the user cannot spend the same electronic coin at two different merchants. Overspending occurs when the same

credential is used more than specified by the usage limit within the same scope. Possible sanctions on overspending could be that the user is simply denied access to the service, or that some further attributes from the user's credential are revealed [CHL06,CHK+06].

With limited-use credentials one can prevent users from sharing and redistributing their credentials to a large extent. Another means of sharing prevention is the so-called all-or-nothing sharing mechanism [CL01]. This mechanism ensures that if a user shares one credential with another user (which requires revealing the other user the secret key material of that credential) then the other user can also use all the other credential (because they are based on the same secret key material). In this case sharing a single credential would mean to share one's whole digital identity, e.g., including access to one's bank account, which people probably are not prepared to do. If, however, one wishes to make sharing of credentials infeasible, then they need to be protected by tamper-resistant hardware, which we discuss next.

Hardware Protection. Being digital, anonymous credentials are easily copied and distributed. One the one hand, this is a threat to verifiers as they cannot be sure whether the person presenting a credential is the one to whom it was issued. On the other hand, this is also a threat to users as it makes their credentials vulnerable to theft, e.g., by malware.

One means to counter these threats is to protect a credential by tamper-resistant hardware device such as a smart cards, i.e., to perform all operations with the credential on the device itself. A straightforward way of doing so in a privacy-friendly way would be to embed the same signing key in all issued smart cards. The disadvantage of this approach is that if the key of one card is compromised, all smart cards have to be revoked.

A more realistic approach is to implement the Camenisch-Lysyanskaya credential system on a standards Java card [BCGS09]. However, depending on the type of smart card, it might only be possible to process a single credential on the device. In this case, one could still bind other credentials to the device by including in each credential an identifier as an attribute that is unique to the user [Cam06]. All of a user's credentials should include the same identifier. (The issuing of these credentials can even be done without having to reveal this identifier.) When an external credential (i.e., a credential that is not embedded in the smart card) is shown, the verifier requires the user to not only show the external credential but also the credential on the smart card, together with a proof that both credentials contain the same identifier as a smart card. Using the attribute properties feature, users can prove that both credentials contain the same identifier without revealing the identifier.

More Privacy-Enhancing Authentication Mechanisms. There are a number of primitives that are related to anonymous credentials. Some of them are special cases of anonymous credentials while others can be seen as building blocks or share the same cryptographic techniques to achieve anonymity.

Blind Signatures. A blind signature scheme [Cha83] allows a user to get a signature from the signer without the signer being aware of the message nor the resulting signatures. Thus, when the signer at some later point is presented with a valid signature on a message, he is not able to link it back to the signing session that produced the signature. Blind signature schemes are a widely used building block for schemes to achieve anonymity. Examples include anonymous electronic voting [Cha83,FOO91] and electronic cash [Cha83], which we discuss below. A large number of different blind signature schemes have been proposed in the literature based on various cryptographic assumptions; there are too many to be listed here.

The main feature of blind signatures that the signer has no control whatsoever on the message being signed. This feature can at the same time be a drawback. Typically, the signer wants to impose certain restrictions on the message that he's signing, such as the expiration date of a credential, or the denomination of a digital coin. When used in protocols, blind signatures therefore often have to be combined with inefficient "cut-and-choose" techniques, where the user prepares many blinded versions of the message to be signed, all but one of which are to be opened again, and the remaining one is used to produce the signature. A more efficient approach is to use *partially* blind signatures [AF96], where the signer determines part of the signed message himself, allowing him to include any type of information, such as the issuance or expiration date of the signature.

Electronic cash. The goal of (anonymous) electronic cash [Cha83] is to prevent fraud while achieving the same privacy guarantees as offered by cash money in the real world. In particular, when a user withdraws and electronic coin from the bank, spends it at a merchant, and the merchant deposits the electronic coin at the bank, the bank cannot link the coin back to the user. However, if either the user or the merchant try to cheat by spending or depositing the same coin twice, the identity of the fraudster is immediately revealed.

Online electronic cash, i.e., where the bank is online at the moment a coin is spent, can be built using blind signatures by having the bank blindly sign random serial numbers. After having issued the blind signature to a user, the bank charges the user's account. The user can spend the money with a merchant by giving away the random serial number and the signature. To deposit the coin, the merchant forwards the serial number and signature to the bank, who verifies the signature and checks whether the serial number has been deposited before. If not, the bank credits the merchant's account; if so, the bank instructs the merchant to decline the transaction.

In off-line electronic cash [CFN88] the bank is not involved when the coin is spent, only when it is withdrawn or deposited. The techniques described above are therefore enhanced to at the time of deposit distinguish between a cheating user and a cheating merchant, and in the former case, to reveal the identity of the cheating user. Both online and off-line electronic anonymous cash can be seen as special cases of limited-use anonymous credentials as described above, where a single scope is used for all payments. To obtain off-line electronic cash,

the user is required to provide a verifiable encryption of her identity, which is only decrypted in case of fraud.

Group Signatures. A group signature scheme [CvH91] allows group members to sign messages in a revocably anonymous way, meaning that any verifier can tell that the message was signed by a group member, but not by which group member, while a dedicated opening manager can lift the anonymity of a signature and reveal the identity of the signer who created it. Group membership is controlled by a central group manager, who generates the group's public key and provides the individual members with their secret signing keys. Some schemes combine the roles of group manager and opening manager in a single entity.

Group signatures satisfy a whole range of security properties, including unforgeability (i.e., no outsider can create valid signatures in name of the group), unlinkability (i.e., signatures by the same signer cannot be linked), anonymity (i.e., nobody except the opening manager can tell which signer created a signature), traceability (i.e., any valid signature can be traced back to a signer), exculpability (i.e., no collusion of cheating signers can create a signature that opens to an honest signer), and non-frameability (i.e., not even a cheating group manager can create a signature that opens to an honest signer). Many of these properties are in fact related [BMW03,BSZ05].

The showing protocol of many anonymous credential systems follows a typical three-move structure that allows them to be easily converted into a signature scheme by means of a hash function [FS87]. The resulting signature scheme inherits all the anonymity features of the credential system. A group signature scheme can then be obtained by combining it with verifiable encryption: the issuer plays the role of group manager and issues to each group member a credential with a single attribute containing its identity. Group members do not reveal their identity attribute when signing a message, but verifiably encrypt it under the public key of the opening manager. One can take this approach even further by including more attributes and using the attribute properties feature. For example, one could create a signature that reveals that some authorized group member between 18 and 25 years old signed the message, but only the opening manager can tell who exactly did.

Ring Signatures. One possible disadvantage of group signatures is that the group manager decides on the composition of the group, and that members can only sign in name of that group. Ring signatures [RST01] are a more flexible variant of group signatures that have no group manager or opening manager. Rather, users can determine the group of "co-signers" at the time a signature is created. The co-signers' collaboration is not needed in the signing process, so in fact, they need not even be aware that they are involved in a ring signature. There is no authority to reveal the identity of the signer behind a ring signature, but some schemes allow the signer to voluntarily prove that they created a signature.

Redactable and Sanitizable Signatures. In some applications it may be necessary to hide words, sentences, or entire paragraphs of a signed document

without invalidating the original signature. Redactable [JMSW02] and sanitizable [ACdMT05] signatures allow exactly that, the difference being that in the former anyone can censor a document, while in the latter only a censoring authority designated by the original signer can do so. Both primitives satisfy a privacy property implying that it is impossible to link back a censored signature to the original signature that was used to create it.

Privacy-Enchancing Encryption. While the main focus of this work is on privacy-enhancing authentication, a complete privacy-friendly infrastructure also involves special encryption mechanisms. We already touched upon verifiable encryption in relation to anonymous credentials. We discuss a selection of other privacy-relevant encryption primitives here.

Anonymous Communication. Most of the anonymous authentication mechanisms described above assume rely on an anonymous underlying communication network: cryptographic unlinkability of signatures clearly does not help if the users are identifiable by their IP address. Mix networks [Cha81] can be used to obfuscate which user communicates with which servers by routing the traffic through an encrypted network of mix nodes. The exact route that a packet follows can either be decided by the mix node or by the sender of the packet. In the latter case, the message is wrapped in several layers of encryption, one layer of which is peeled off at each node; this process is often referred to as onion routing [Cha81,GRS99,CL05]. So-called dining cryptographer networks or DC-nets [Cha88] even hide the fact whether entities are communicating at all, but they of course incur a constant stream of dummy traffic between all participants in doing so.

Homomorphic and Searchable Encryption. With current technology trends such as software as a service and cloud computing, more of our information is stored by external services. Storing the information in encrypted form is often not an option, as it ruins either the service's functionality or its business model. As the main goal of encryption is to hide the plaintext, it usually destroys any structure present in the plaintext; tampering with a ciphertext either renders it invalid, or turns the plaintext into unpredictable random garbage. Some encryption algorithms however are homomorphic, in the sense that applying certain operations on ciphertexts has the effect of applying other operations on the plaintexts. One can thereby process encrypted data without decrypting it, so that for example a server can apply data mining mechanisms directly on encrypted information [OS07]. There exist homomorphic encryption schemes that support multiplication [ElG85] and addition [Pai99] of plaintexts, and since recently, also schemes that support both at the same time [Gen09].

In similar scenarios it can be useful if a server can search through encrypted information without having to decrypt it. For example, this would enable an encrypted email hosting server to perform efficient searches on your email and transmit only the matching (encrypted) emails. Special-purpose schemes have

been developed for this purpose as well, both in the symmetric [SWP00] and the asymmetric [BCOP04] setting.

Oblivious Transfer. Imagine a database containing valuable information that is not sold as a whole, but that rather charges customers per accessed record. At the same time, the list of queried records reveals sensitive information about the customers' intentions. For example, a company's search queries to a patent database or to a DNA genome database may reveal its research strategy or future product plans.

An oblivious transfer protocol [Rab81] solves this apparently deadlocked situation by letting a client and server interact in such a way that the server does not learn anything about which record the client obtained, while the client can only learn the content of a single record. The adaptive variant [NP99] of the primitive can amortize communication and computation costs over multiple queries on the same database.

2.2 Example Applications

In this section we give examples of privacy-sensitive applications for which protocols have been developed by combining some of the tools we just discussed.

Electronic Voting, Polling, and Petitions. Voting privacy is more than just a desirable feature, it is a fundamental principle for a democratic election. Electronic voting schemes have been proposed based on mix networks [Cha81], based on homomorphic encryption [CF85], and based on blind signatures [FOO92].

Direct Anonymous Attestation. How can a verifier check that a remote user is indeed using a trusted hardware module, without infringing on the privacy of the user, and without having to embed the same secret key in each module? This questions arose in the context of the Trusted Computing Group (TCG). In particular, the Trusted Platform Module (TPM) monitors the operating system and then can attest to a verifier that is pristine, e.g., free of viruses and thus safe to run an application such as e-banking. To protect privacy, the TCG has specified a scheme for this attestation that can essentially be seen as a group signature scheme without the opening functionality, so that anonymity cannot be revoked [BCC04] but with a revocation feature such that stolen keys can nevertheless be identified and rejected.

Oblivious Transfer with Access Control and Prices. The techniques described above can be combined in various way to address interesting business needs. For example, imagine that each record in a patent or DNA database as described above is protected by a different access control policy, describing the roles or attributes that a user needs to have in order to obtain it. By combining anonymous credentials with adaptive oblivious transfer protocols, solutions exist where the user can obtain the records she's entitled to, without revealing the applicable

access control policy to the database, or which roles she has [CDN09]. By another combination of such techniques, the database can attach different prices for each record, and let users only download as many records as their prepaid balance allows, all while remaining completely anonymous [CDN10].

3 Conclusion

Even though a large number of very advanced privacy-enhancing cryptographic primitives have been proposed in the literature, their way to broad-scale deployment in the real world presents still a number of challenges.

One is the design of user interfaces that capture the core concepts of the underlying cryptography, while hiding the details.

Another challenge is the integration of the cryptographic primitives in the overall (authentication and access control) infrastructure. For instance, to deploy anonymous credentials, one needs proper policy languages to express and communicate the access control requirements in a way that supports, e.g., selective revealing of attributes, or proving properties of attributes. Too often do such languages implicitly assume that the user reveals all of her attributes by default. Moreover, since credential attributes are often sensitive information, these policy languages have to be integrated with privacy policy languages in which servers can express how the revealed information will be treated, and for users to express to whom and under which circumstances they are willing to reveal it. Privacy policy languages such as P3P [W3C06] are a first step, but are often not fine-grained enough, and lack the tight integration with access control policies. These and other challenges are currently being addressed as part of the PrimeLife project [prib,CMN+10].

From a cryptographic perspective there are still many open problems to be addressed. Researchers are searching for more efficient primitives, as in many applications the incurred overhead is still prohibitive. Also, dedicated protocols for advanced applications like social networks or location-based services would be desirable. From a theoretical point of view, an important challenge is how existing primitives can be securely and efficiently composed to build new, more complex primitives. Finally, most of the above primitives currently still lack proper key management infrastructures so that keys can be securely stored, authenticated, and revoked.

Acknowledgements

The authors enjoyed many exciting discussions with the participants of the PrimeLife project, some of them leading us to write this overview. Thank you all! This work was supported in part by the European Community through the Seventh Framework Programme (FP7/2007-2013) project PrimeLife (grant agreement no. 216483).

References

[ACdMT05] Giuseppe Ateniese, Daniel H. Chou, Breno de Medeiros, and Gene Tsudik. Sanitizable signatures. In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, *Computer Security - ESORICS 2005*, volume 3679 of *Lecture Notes in Computer Science*, pages 159–177. Springer, 2005.

[AF96] Masayuki Abe and Eiichiro Fujisaki. How to date blind signatures. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology – ASIACRYPT '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 244–251. Springer, 1996.

[BCC04] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *Proc. 11th ACM Conference on Computer and Communications Security*, pages 225–234. acm press, 2004.

[BCGS09] Patrik Bichsel, Jan Camenisch, Thomas Groß, and Victor Shoup. Anonymous credentials on a standard Java Card. In to appear, editor, *ACM Conference on Computer and Communications Security*, 2009.

[BCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer, 2004.

[BDD07] Stefan Brands, Liesje Demuynck, and Bart De Decker. A practical system for globally revoking the unlinkable pseudonyms of unknown users. In Josef Pieprzyk, Hossein Ghodosi, and Ed Dawson, editors, *ACISP*, volume 4586 of *Lecture Notes in Computer Science*, pages 400–415. Springer, 2007.

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629. Springer Verlag, 2003.

[Bra99] Stefan Brands. *Rethinking Public Key Infrastructure and Digital Certificates— Building in Privacy.* PhD thesis, Eindhoven Institute of Technology, Eindhoven, The Netherlands, 1999.

[BS04] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel, editors, *ACM CCS 2004*, pages 168–177. ACM, 2004.

[BSZ05] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In Alfred Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 136–153. Springer, 2005.

[Cam06] Jan Camenisch. Protecting (anonymous) credentials with the trusted computing group's tpm v1.2. In Simone Fischer-Hübner, Kai Rannenberg, Louise Yngström, and Stefan Lindskog, editors, *SEC*, volume 201 of *IFIP*, pages 135–147. Springer, 2006.

[CD00] Jan Camenisch and Ivan Damgård. Verifiable encryption, group encryption, and their applications to group signatures and signature sharing schemes. In Tatsuaki Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 331–345. Springer Verlag, 2000.

[CDN09] Jan Camenisch, Maria Dubovitskaya, and Gregory Neven. Oblivious transfer with access control. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM Conference on Computer and Communications Security*, pages 131–140. ACM, 2009.

[CDN10] Jan Camenisch, Maria Dubovitskaya, and Gregory Neven. Unlinkable priced oblivious transfer with rechargeable wallets. In *Financial Cryptography 2010*, 2010.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo G. Desmedt, editor, *Advances in Cryptology — CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer Verlag, 1994.

[CF85] Josh D. Cohen and Michael J. Fischer. A robust and verifiable cryptographically secure election scheme (extended abstract). In *FOCS 1985*, pages 372–382. IEEE, 1985.

[CFN88] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In Shafi Goldwasser, editor, *Advances in Cryptology – CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 319–327. Springer, 1988.

[Cha81] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, February 1981.

[Cha83] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology — Proceedings of CRYPTO '82*, pages 199–203. Plenum Press, 1983.

[Cha88] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[CHK+06] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. In *ACM Conference on Computer and Communications Security*, pages 201–210, 2006.

[CHL06] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Balancing accountability and privacy using e-cash (extended abstract). In *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 141–155, 2006.

[CKS09] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography*, pages 481–500, 2009.

[CL01] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer Verlag, 2001.

[CL02] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76. Springer Verlag, 2002.

[CL05] Jan Camenisch and Anna Lysyanskaya. A formal treatment of onion routing. In Victor Shoup, editor, *Advances in Cryptology — CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 169–187. Springer Verlag, 2005.

[CMN+10] Jan Camenisch, Sebastian Mödersheim, Gregory Neven, Franz-Stefan Preiss, and Dieter Sommer. A language enabling privacy-preserving access control. In *To appear at SACMAT 2010*. ACM, 2010.

[CS03] Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In Dan Boneh, editor, *Advances in Cryptology — CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 126–144, 2003.

[CSF+08] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (Proposed Standard), May 2008.

[CvH91] David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *Advances in Cryptology — EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer-Verlag, 1991.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In George Robert Blakley and David Chaum, editors, *Advances in Cryptology — CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer Verlag, 1985.

[FOO91] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. Interactive bi-proof systems and undeniable signature schemes. In Donald W. Davies, editor, *Advances in Cryptology — EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 243–256. Springer-Verlag, 1991.

[FOO92] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In Jennifer Seberry and Yuliang Zheng, editors, *Advances in Cryptology - AUSCRYPT '92*, volume 718 of *Lecture Notes in Computer Science*, pages 244–251. Springer, 1992.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer Verlag, 1987.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing (STOC 2009)*, pages 169–178. ACM, 2009.

[GRS99] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42(2):84–88, February 1999.

[JMSW02] Robert Johnson, David Molnar, Dawn Xiaodong Song, and David Wagner. Homomorphic signature schemes. In Bart Preneel, editor, *Topics in Cryptology - CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 244–262. Springer, 2002.

[NFHF09] Toru Nakanishi, Hiroki Fujii, Yuta Hira, and Nobuo Funabiki. Revocable group signature schemes with constant costs for signing and verifying. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography*, volume 5443 of *Lecture Notes in Computer Science*, pages 463–480. Springer, 2009.

[NP99] Moni Naor and Benny Pinkas. Oblivious transfer with adaptive queries. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 573–590. Springer, 1999.

[OS07] Rafail Ostrovsky and William E. Skeith III. Private searching on streaming data. *Journal of Cryptology*, 20(4):397–430, 2007.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite residuosity classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–239. Springer Verlag, 1999.

[pria] PRIME project, website. www.prime-project.eu.

[prib] PrimeLife project, website. www.primelife.eu.

[Rab81] Michael O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.

[SWP00] Dawn Xiaodong Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *IEEE Symposium on Security and Privacy*, pages 44–55, 2000.

[W3C06] W3C. The Platform for Privacy Preferences 1.1 (P3P1.1) Specification, 2006. http://www.w3.org/TR/P3P11/.