Interoperation between a conventional PKI and an ID-based infrastructure

Geraint Price and Chris J. Mitchell

Information Security Group, Royal Holloway, University of London,
Egham, Surrey TW20 0EX, UK
{Geraint.Price, C.Mitchell}@rhul.ac.uk

Abstract. In this paper we consider how practical interoperation between a conventional PKI and an infrastructure based on ID-based cryptography might be achieved. Major issues arising from such interoperation are raised, and possible solutions are proposed.

1 Introduction

Suppose a domain (domain A) uses a 'conventional' public key infrastructure (PKI), with one or more Certification Authorities (CAs). Suppose also that a second domain (domain B) operates an ID-based public key infrastructure, with one or more trusted key generation entities. Now suppose that the members of the two domains wish to inter-operate, i.e. verify signatures generated by each other and/or send each other encrypted messages. This clearly presents a variety of problems, and it is the purpose of this paper to look at these issues.

Of course, one pre-requisite for such a scheme to work is that the members of domain A must support the ID-based crypto-algorithms used by members of domain B, and vice versa. However, this should be relatively straightforward to achieve either by pre-configuring all entities to support a sufficiently wide variety of algorithms, or by using mechanisms such as signed applets, etc.

The main focus of this paper is to address the underlying key management issues, together with the policies and practices that need to be put in place to support interoperation. That is, we investigate the management issues underlying the interoperation of a certificate-based PKI with an Identity-based one.

In particular we consider issues such as:

- What are the specific issues arising for interoperation in either direction?
- What are the security implications of interoperation?
- What lessons can we learn, and where do we go from here?

Essentially, understanding how to support interoperation comes down to understanding the differences in operation between the two types of infrastructure, and how this affects the security architecture.

The remainder of this paper is structured as follows. Section 2 gives an overview of public key and ID-based cryptography, highlighting the differences between certificate based infrastructures and identity-based infrastructures. This includes a brief review of the (limited) prior art in the field. This leads naturally into Section 3, where we provide a description of a certificate-based solution to the interoperation problem. This is broken down into three subsections: an overview of inter-domain operation in 'native' environments; how to get a ID-based solution to work for the certificate-based user; and how to get a certificate-based solution to work for the ID-based user. Section 4 moves on to look beyond the use of certificates to solve the interoperation problem, highlighting both some potential alternative solutions and other issues that might come into play (e.g. the use of the Al-Riyami-Paterson certificateless public key infrastructure schemes [1, 2]). In Section 5 we provide an overview of the lessons learnt to date, and we outline possible future work. Finally, Section 6 gives some conclusions.

2 Technical Background

In this section we briefly review the technical background which we require for our subsequent analysis. We begin by reviewing the history of Identity-based (or Identifier-based) public key cryptography, outlining the differences between it and certificate-based infrastructures. We then close this section by reviewing what prior art exists discussing possible interfaces between certificate-based and identifier-based cryptographic infrastructures.

In practice, the management, distribution and use of public key certificates as part of a conventional PKI adds significant overheads. It was with this drawback in mind that, in 1984, Shamir proposed a new form of public key cryptography [13] which he termed Identity-based cryptography. In the 'traditional' model of public key cryptography, both the public and the corresponding private key are random in appearance; by contrast, in the ID-based model, the public key pair is generated from publicly verifiable data (with the generation of the corresponding private key requiring an additional secret parameter held by a Trusted Authority). Shamir pointed out that, if the public key is generated from an identifier of the entity to whom the key is assigned (e.g. a user's email address), this would obviate the need for certificates, and would simplify the management of public keys in a fielded system.

Until relatively recently the main limitation of ID-based cryptography was that, while Shamir and many others were able to propose a variety of ID-based signature schemes (see, for example, [11]), no efficient ID-based encryption scheme was known. This situation changed in 2001, when Boneh and Franklin [3, 4] proposed a practical and efficient encryption scheme.

The main benefit from the use of ID-based encryption, as discussed in [13], is the ability to encrypt a message sent to a given recipient without first having to retrieve the recipient's public key. All that is required is knowledge of the recipient's identity. By contrast, the potential practical benefits of using an identity-based signature scheme are rather less. This is because, for signatures, the private key needs to be created before a signature can be created in both the conventional and the ID-based cases. This meant that, until Boneh and Franklin's discovery, very little work had been conducted on the practical use of ID-based cryptography; however, since then, there has been a resurgence of interest in the field.

We next briefly discuss some of the main operational differences. Firstly, observe that, for ID-based encryption, the private key only needs to be generated when it is actually needed to decrypt data. Secondly, while there is no need for certificate revocation *per se* within the system, controlling the lifetime of valid identifiers results in very similar implementation problems to those encountered in a conventional PKI, as discussed by Paterson and Price [12].

Whilst this paper discusses the possible interaction between a traditional certificate-based public key infrastructure and an identifier-based infrastructure, there is almost nothing in the published literature that looks at this issue. We now briefly review relevant previous work.

Chen et al. [5] discuss the use of a hybrid scheme, with a traditional certificate-based PKI being used by the Trusted Authority (TA) at the domain/organisational level, but with identity-based cryptography used at the user level. They base their design on the premise that certificate-based systems are more efficient at the domain level and are less efficient at the user level, but that the inverse is true for identity-based systems. They note that they require cross-certification between the domain trust authorities, but do not provide detailed discussions of this issue.

Smetters and Durfee [14] propose an extension to DNSSEC, which is itself a recently proposed secure extension of DNS. They discuss the problems that would be faced if a global identity-based system were to be proposed. The main drawback they highlight would be the need for a central TA, and the management of a hierarchy of TAs under a root TA. In the case of identity-based systems, the keys of the mid-level TAs could be generated by TAs further up the tree, which might be an undesirable property. They propose that each individual trust domain runs its own TA, with their system parameters being distributed via DNSSEC.

While this prior work does consider the problems of trying to implement an identity-based scheme on a large scale, both proposals offer a tiered approach at a global scale, where it is assumed that peer-level communication is carried out using the same cryptographic mechanisms. In the remainder of this paper we analyse the technical challenges that would arise if users in two different organisations using different underlying mechanisms were to wish to communicate securely.

3 A Simple Certificate-based Solution

In this section we consider a simple certificate-based solution to the technical challenge faced when users in domains with different types of asymmetric infrastructures wish to communicate with each other. The basic outline of the design is to have the two separate domain infrastructures (one certificate-based and one identity-based) use traditional cross-certificates to authenticate the CA to the identity-based domain, and the TA to the certificate-based domain.

We divide the discussion into three main parts. In Section 3.1 we consider what would happen if domains of the same type (i.e. certificate-to-certificate or identity-to-identity) need to interoperate. Then in section 3.2 we review our cross-certification design from a certificate-based user's perspective, and in section 3.3 we review our design from an identity-based user's perspective.

3.1 Core principles

Before detailing our proposed solution to certificate-to-identity cross-domain communication, we review how cross-domain communication happens in native environments.

Certificate-based : The traditional method of supporting inter-domain communication between multiple certificate-based environments is to use *cross-certificates*. Cross-certificates are certificates issued by a CA to certify a peer-level CA's root public keys. This allows the client software of users in one domain to validate the certificates of users within another domain by first verifying the cross-certificate signed by their own CA (or, more generally, by verifying a chain of cross-certificates). While this solution is relatively straightforward, it has problems. Most notably, these problems arise when the two CAs reside in logically separate security domains. As a result, certificate path discovery can sometimes be complicated. Even if the certificate path can be built, the analysis of Certificate Policies (CP) and Certification Practice Statements (CPS) [6] can make it difficult for the validation mechanism of a user in one domain to know whether the certificate they have is suitable for the task which the user wishes to use it for.

ID-based : Although very little work has been done on inter-operation between identity-based domains, both Chen et al. [5] and Smetters and Durfee [14] propose the use of security mechanisms 'external' to identity-based cryptography to authenticate cross-domain credentials. Chen et al. use traditional certificate-based mechanisms, and Smetters and Durfee use DNSSEC. It seems apparent that identity-based cryptography itself is not suitable for supporting the authentication of inter-domain credentials. This view would appear to be validated by the work of Chen et al. and Smetters and Durfee on cross-domain communication.

Similar problems to those encountered in certificate-based interaction arise here, notably that the policies operated within the two separate domains might need to be analysed for compatibility. However, at least in the case of identity-based encryption (as opposed to signature), path discovery might be simpler as the recipient could be forced to retrieve a new private key from their local TA. This would mean that the sender essentially offloads path discovery to the TA in the recipient's domain.

As we see below, the problems highlighted here are likely to get worse as the two different types of domains interact. This is mainly due to the fact that certificate-based users need to be able to make use of, and make sense of, ID-based parameters, and vice versa. By necessity, this adds a layer of complexity to the system architecture.

3.2 The certificate-based user

For the reminder of this paper we consider two security domains. We suppose domain **A** runs a traditional certificate-based PKI, while domain **B** runs an identity-based infrastructure.

From our discussions in the previous section we note that using a cross-certificate would appear to be the logical choice for supporting peer-entity authentication between the CA and TA. In this section we look at the design of such a scheme from a certificate-user's point of view.

We start by providing a simplified overview of how the design would work.

- The CA for domain **A** generates a cross-certificate for the TA in domain **B**, containing the TA's public parameters.
- This cross-certificate could contain policy information for domain **B** of relevance to a user in domain **A**.
- A user in domain **A** validates the cross-certificate and checks the policy for acceptability (we discuss how this policy checking might be carried out below).
- As the certificate is likely to be obtained from a local repository, there will typically be a need for a certificate status check, and possibly some additional path validation, to be carried out on the certificate.

The idea of using a certificate to authenticate the public parameters of a TA is not new. Apocryphal evidence suggests that some organisations looking to implement identity-based encryption in practice intend to use a traditional PKI certificate to allow users within the system to verify the domain parameters for an ID-based system.

The above technique would allow members of domain **A** to derive public keys from entities within domain **B**. Hence, this would enable them to verify signed message from, and send encrypted messages to, members of domain **B**.

We now discuss some of the operational details of such a scheme. In doing so, we note some potential difficulties that arise from the fundamental differences between certificate and identity based infrastructures.

Certificate Type and Content The first area we explore is the issue of which type of certificate to use for the cross-certificate. That is, should it be a traditional X.509 identity certificate or an X.509 attribute certificate. As well as the choice of certificate type, we also consider the contents of such a cross-certificate.

There are potential advantages and disadvantages associated with the use of both types of certificate. The arguments can be summarised as follows.

Identity Certificate : This would appear to be the more logical choice, with the identity of domain **B** being bound to the scheme parameters used within **B**'s identity-based implementation. This could be achieved by defining new extension fields for the X.509 certificate format to manage this additional information. This approach would provide the user in domain **A** with a mechanism that most accurately reflects the way in which cross-certification is carried out in a certificate-based environment. However, it does come at a cost. Because of the very nature of identity-based schemes, there is no explicit means of revocation. One way of dealing with this problem is to periodically update the scheme parameters within domain **B** in order to refresh all keys in the system. This would increase the pressure on the certificate management procedures of the cross-certificate for domain **A**'s CA. This cross-certification could be made more lightweight by not including the scheme parameters in the certificate. Thus, we would be relying on the certification of the domain **B**'s identity alone.

Attribute Certificate : An alternative would be to use an attribute certificate to certify domain **B**'s TA's right to produce valid system parameters. This effectively extends domain **A**'s security associations to domain **B**, purely for the act of managing the identity-based parameters. This solution is more lightweight from domain **A**'s viewpoint, as it is up to the TA in domain **B** to update the system parameters and authenticate them within domain **A**. However, we believe that an attribute certificate will provide a weaker binding between the cross-certificate and the system parameters, as the CA in domain **A** is only certifying the identity of domain **B**, and not the system parameters directly. As well as the weaker binding, such a scheme would probably require the TA in domain **B** to maintain an additional set of certificate-based keys in order to authenticate the system parameters to the users in domain **A**. There is also the question as to whether the TA in domain **B** really has the 'right' to carry out an action in domain **A**. We believe that further discussions on this point could help resolve how the powers of domain **B**'s TA could be expressed within domain **A**'s security policy. However, this mechanism could be strengthened by having the TA's system parameters included within the certificate as an additional set of attributes.

As we can see from our discussion, both constructs have their benefits and drawbacks. Additionally, we have noted that both types of certificate can be modified to present a tighter or more lightweight binding within each certificate type.

We believe that the use of an X.509 identity certificate would be preferable in practice, with the system parameters for domain **B** included either as the public key content, or in a special extension field. This would also more naturally map the way in which a user in domain **A** might manage existing cross-certificates to certificate-based domains. Also, an identity certificate is traditionally associated with the provision of authentication, which more accurately captures the nature of cross-certification in this environment.

CPs and CPSs The biggest difference between certificate-based infrastructures and identity-based infrastructures is in the way they handle policies. In a certificate-based PKI, great reliance is placed on the use of Certificate Policies and Certification Practice Statements [6]. Conversely, much is made within an identity-based infrastructure of off-loading policy creation to the end user, while centralising policy enforcement. An example of this is the NHS trial for identity-based encryption which allows the sending party to encode the policy as the key [7][1].

In the case of a user in domain **A** sending an encrypted message to a user in domain **B**, which policy should the sender follow? If the identifier-based domain does not have the equivalent of the Object Identifiers (OIDs) in X.509 certificates, then how does the user in domain **A** know that the rules used in assigning private keys in domain **B** will conform to the desired policy?

We present two ways in which this problem could be solved.

- The TA in domain **B** could release a set of policy statements in a similar manner to a CP created within domain **A**. One of the stated benefits of ID-based cryptography is the ability for users within an ID-based domain to generate their own policies on the fly which can subsequently be verified at the TA before it issues the associated private key. In such cases, these client-generated policies are likely to be checked against more coarse-grained policies already held at the TA. We believe that such coarse-grained policies are likely to need refinement before being published — as in a CP. These policy statements could then be included in the cross-domain certificate, and hence can be checked by a user in domain **A**. While this would simplify the work for the user in domain **A**, it complicates the policy management within domain **B**.
- The sender within domain **A** could generate a new identity-based public key for the recipient, making part of the encoding a reference to the policy for the decryption and sending it with the message. The recipient in domain **B** would then need to go to their TA to retrieve the new decryption key. This would complicate the task for the sender, but would maintain the ability within domain **B** for the TA to validate the decryption request just prior to the decryption.

As can be seen from the above discussion, there are benefits and drawbacks to both mechanisms. We believe that the application requirements are likely to be a determining factor in deciding which mechanism to use in practice.

3.3 The ID-based user

We next review the use of cross-certificates within the identity-based user's domain. A simplified overview of how the design would work is as follows.

[1] In this case the terminology changes slightly to that of identifier-based encryption, where the publicly verifiable information is an identifier that is used to encode the policy statement.

- The TA for domain **B** generates a cross-certificate for the public key of the CA in domain **A**.
- This cross-certificate could contain policy information for domain **A** of relevance to a user in domain **B**.
- A user in domain **B** validates the cross-certificate and checks the policy for acceptability.
- As the certificate is likely to be obtained from a local repository, there will typically be a need for a certificate status check, and possibly some additional path validation, to be carried out on the certificate.

This mirrors the design for the certificate-based user, as discussed in section 3.2. While this would give us the functionality we require for inter-domain interaction, we believe that it is a relatively inelegant solution. The main drawback for the users in domain **B** is that, before making use of the recipient's public key, they will be required to interact with another entity to fetch the certificate for the user in domain **A**. This means that one of the key advantages of using ID-based encryption is lost.

An alternative would be to have the TA in domain **B** issue an identity-based signing key to the CA within domain **A**. The CA could then dual-sign the certificates for its users, such that a user in domain **B** could validate the signatures directly using domain **B**'s system parameters. However, we believe that this approach is unlikely to be used in practice. The main reason for this is that, because of the very nature of identity-based cryptography, a signing key issued by the TA is escrowable (unless an approach such as CL-PKC, as outlined in Section 4.2, is used). Thus, a corrupt TA in domain **B** could falsify certificates or messages from the CA in domain **A**. The CA is unlikely to want to put itself in a situation where this is possible.

We now look at the certificate content, policy issues, and revocation from the perspective of an identity-based user.

Certificate Content If we assume that the certificate is an X.509 cross-certificate, then the content is relatively well defined. The TA in domain **B** would, however, need to generate and use a different type of signature key in order to generate the certificate; this adds complexity to the scheme. However, the user in domain **B** would be expected to make use of the user certificates within domain **A** in order to enable secure communication, and thus it is not a significant additional burden.

CPs and CPSs As we saw from our discussion in the previous section, the differences in the ways policies are managed in the two systems is one of the main hurdles to integration. The lack of prior work on CP/CPS equivalents in the identity-based domain would appear to complicate matters here.

Two candidate approaches for dealing with this problem can be identified.

- The users in domain **B** could be required to parse the certificate policies from domain **A** directly.

- The TA within domain **B** could parse and identify suitable CPs from domain **A**. The TA could then make available a list of the CPs which it deems acceptable for use within domain **B**.

The second option provides a much cleaner approach. Not only does it reduce the complexity of interactions at the user level, but it more accurately reflects the existing balance of policy checking in an identity-based environment. The TA is able to regulate the issuing of private keys on a more ad hoc basis, with some schemes requiring policy identifiers to be included within the public key to specify the policy that must be enforced when using that key pair [7]. This contrasts with the way in which clients in a certificate-based domain are expected to download the CPs of relevance and make local decisions before making use of a public key for encryption.

Revocation Possibly the largest hurdle to interaction is the lack of explicit revocation within an identity-based domain. A ‘pure’ identity-based domain does not have explicit key revocation, but is more likely to rely on key re-issuance. If a user in domain **B** wishes to use a key for a user in domain **A**, then they are likely to need to validate the current certificate status. There are the three ways in which we believe this could be achieved.

- The CA could issue new certificates at the same rate as the TA would issue identity-based public keys for its users.
- The TA could act as a filter for user requests, where the TA performs the revocation interrogation on behalf of its clients.
- The user could be required to interrogate and interpret CRLs or OCSP servers directly.

The first two solutions are cleaner for users in domain **B**. The first solution is, however, likely to be too computationally intensive, as certificate distribution in a traditional PKI often incurs a large overhead. The second solution could result in the TA becoming a resource bottleneck, but fits closer to the identity-based model, where the TA is given the power to control key access within a secure environment. The third solution is simplest from an architectural perspective, but it increases the complexity of certificate processing at the user level. In practice, the application level considerations are clearly going to impact on this design decision.

4 Extending the Analysis

In this section, we briefly review some potential alternatives to the certificate-based solution presented above, along with additional technologies which could impact on an inter-domain design.

4.1 Potential Alternative Solutions

A high level outline of some possible alternative designs for inter-domain interoperability is as follows.

- The design of the scheme presented in Section 3 requires the sender of an encrypted message to do the majority of the additional work. It would be possible to reverse this burden. For example, an identity-based user could act as though the recipient is "local" to the identity-based domain, and force the certificate-based user to retrieve the necessary identity-based private key from the TA in domain **B**.
- A trusted intermediary could be set up to act as a server that decrypts, then re-encrypts, the flow of messages. This would have no major impact in the identity-based domain, as the TA can already read anything that is sent to its clients (again, unless a scheme such as CL-PKC is used — see below). However, this might contravene the security policy in the certificate-based domain, where key or plaintext escrow is only carried out in exceptional circumstances.
- In the case of signatures, alternative solutions could be built that make use of the flow of signed messages. Much of the complexity introduced in the previous sections is designed to deal with the issue of retrieval of public encryption keys that is necessary before encryption can take place. However, because a signature has to be generated before any verification can take place, the verification key can be bound to the document or message being signed. This can greatly simplify the key retrieval protocol.

4.2 Additional Technologies

In this section we briefly outline some additional technologies which could impact on any cross-infrastructure design.

Certificateless Public Key Cryptography One of the limitations of identity-based cryptography is the fact that the TA has the ability to escrow all private keys used in the system. Al-Riyami and Paterson developed Certificateless Public Key Cryptography (CL-PKC) [1, 2] in order to overcome this problem (see also a variety of related work, including the notion of 'self-certified public keys', due to Girault [10], and Gentry's 'certificate-based encryption' [8]). The notion of CL-PKC gives the benefits of an identity-based public key mechanism, where the key pair is derived from publicly identifiable information. However, the private key is created in a joint process between the TA and the user, where the TA only knows a partial share of the resulting completed private key. This circumvents the key escrow problem.

Using a CL-PKC mechanism within the identity-based environment would address some of the problems we discuss in the previous sections, most notably as follows.

- It would allow the TA in domain **B** to give a signing key to the CA in domain **A**. This would simplify the parsing of the cross-certificates for the users in domain **B**.
- It would mean that private keys within domain **B** are more likely to fit the policies of use within domain **A**, given that most traditional PKIs completely reject the escrow of signing keys, and only carry out escrow of decryption keys in strictly controlled circumstances.

Certificate Chaining Certificate chaining is a fundamentally important technique where multiple CAs are employed in certificate-based environments. Related work in the identity-based literature aims at generating tiered hierarchies of identity-based keys [9]. However, we believe that most keys within an identity-based domain are likely to be issued directly by the TA in a flat hierarchy within the domain. This would appear to make the best use of the control of private key issue inherent to identity-based cryptography.

The obvious question that results is to ask whether one could imagine a chain which consists of a mixture of certificates signed by regular CAs and keys/certificates generated by TAs? [2] If this were to occur it could potentially be both a benefit and drawback to the way in which such a chain would be processed.

- It would be of benefit if almost all of the keys in domain **B** are "grounded" at the TA. This would simplify the chain reduction algorithms when a client in a domain **A** is assessing a key from the identity-based domain.
- It would present a problem for users in domain **B** who are required to assess a certificate chain, as they are unlikely to be used to processing chains in an identity-based environment.

We note that both these assertions depend on how identity-based mechanisms are used in practice. It will thus be interesting to return to this question when more real-world examples exist.

5 Lessons Learnt and Future Work

In this section we highlight the early lessons we have learnt from this ongoing research. We also identify the avenues that we need to explore further in order to gain a better understanding of how the two infrastructures differ, and how that would impact on any potential interoperation.

- What are the main differences between the two infrastructure types which we have assessed to date?

[2] We note that, in the scenario explored in Section 3, the cross-certificate from domain **B** to domain **A** and the user certificate signed by the CA already constitute a chain.

    - We believe that the main differences lie in how the policy setting and validation is performed. Ensuring that the CP/CPS can be adequately expressed in an identity-based infrastructure will present a major challenge. Conversely, it is important to ensure that the CP/CPS of the certificate-based domain is adhered to by the TA when issuing keys that will be used in inter-domain communication. This work could require the development of a form of policy matching algorithm for the two domains.
    - Where and how the policy matching takes place can impact upon the implementation of that policy. In a certificate-based environment the policy is verified by the user before making use of the key. By contrast, in an identity-based environment the policy is often verified by the TA immediately prior to the release of the private key. Taking these differences into account can be important when designing a security infrastructure.
    - It is important to understand the difference between what is being certified in a certificate-based environment, and what is built into an identifier in an identity-based environment (i.e. does the identity/identifier content correspond to the content of an X.509 certificate?).
- What additional work do we need for our analysis?
    - A clearer assessment and understanding of the differences in policy handling in light of actual application security requirements is needed. This can only come from genuine practical experience in rolling out ID-based infrastructures.
    - Although we only briefly discussed the potential use of attribute certificates in cross-certification, we believe that further research in this area is needed. One of the proposed strengths of identity-based cryptography is its ability to build authorisation policies and implement them directly into the key management infrastructure (see, for example, the trials carried out with the NHS in the UK [7]). Comparing how such use of identity-based cryptography might fit with attribute certificates would seem to be a logical next step.
- What shape should any further analysis take? Our next step will be to develop some example scenarios to provide us with a more detailed understanding of how the outstanding issues might be resolved.

Based on these early results, we believe that the difficulties we highlight above, combined with the technical solutions required, point us towards future use of a form of hybrid architecture, if interoperation is to be viable.

6 Conclusions

In this paper we have analysed a potential means of interoperation between a traditional certificate-based infrastructure and identity-based one.

Our main conclusion is that the existing technical solutions are far from ideal for the users in identity-based environments. The two main reasons for this are: the solutions either force the identity-based client to make use of certificates (bringing with it all the associated problems); or they require the use of trusted intermediates (e.g. trusted decryption servers, Delegated Path Validation-like services). Both of these solutions move us away from the benefits of identity-based cryptography.

In addition, it would appear that building mechanisms to reduce the impact of interoperation at the user level, forces us down a similar route to the proposals for path discovery algorithms in certificate-based environment. For example, it is likely that services would need to be set up to convert CPs in the certificate-domain to policy-centred identifier-based keys in the identity-based domain.

However, we close our discussion by highlighting the fact that interoperation between any security infrastructures can pose major headaches, and the difficulties we have highlighted in this paper are common problems.

Acknowledgements

The authors would like to thank Prof. David Chadwick and various anonymous referees for their contribution in helping clarify the presentation of some of the arguments within this paper.

References

1. S. S. Al-Riyami and K. G. Paterson. Certificateless public key cryptography. In C. S. Laih, editor, *Advances in Cryptology — ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 452–473. Springer-Verlag, Berlin, 2003.
2. S. S. Al-Riyami and K. G. Paterson. CBE from CL-PKE: a generic construction and efficient schemes. In S. Vaudenay, editor, *Public Key Cryptography — PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*, pages 398–415. Springer-Verlag, Berlin, 2005.
3. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer-Verlag, 2001.
4. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Computing*, 32:586–615, 2003.
5. L. Chen, K. Harrison, A. Moss, D. Soldera, and N.P. Smart. Certification of public keys within an identity based system. In A. H. Chan and V. D. Gligor, editors, *Information Security, 5th International Conference, ISC*, volume 2433 of *LNCS*, pages 322–333. Springer-Verlag, 2002.
6. S. Chokhani and W. Ford. RFC 2527: Internet X.509 public key infrastructure certificate policy and certification practices framework, March 1999.
7. Chris R. Dalton. The NHS as a proving ground for cryptosystems. Technical Report HPL-2003-203, Trusted Systems Laboratory, HP Laboratories, Bristol, October 2003.
8. C. Gentry. Certificate-based encryption and the certificate revocation problem. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 272–293. Springer-Verlag, 2003.
9. C. Gentry and A. Silverberg. Heirarchical ID-based cryptography. In Y. Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer-Verlag, 2002.

10. M. Girault. Self-certified public keys. In D. W. Davies, editor, *Advances in Cryptology — EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 490–497. Springer-Verlag, 1992.
11. A. Menezes, P. C. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, 1997.
12. K. G. Paterson and G. Price. A comparison between traditional public key infrastructures and identity-based cryptography. *Information Security Technical Report*, 8:57–72, 2003.
13. A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology – CRYPTO'84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1984.
14. D. K. Smetters and G. Durfee. Domain-Based Administration of Identity-Based Cryptosystems for Secure Email and IPSEC. In *Proceedings 12th USENIX Security Symposium*, pages 215–229, 2003.