Provably Secure Implicit Certificate Schemes

D.R.L. Brown[1], R. Gallant[1], and S.A. Vanstone[1,2]

[1] Certicom Research, Canada
{dbrown,rgallant,svanstone}@certicom.com
[2] University of Waterloo, Canada

Abstract. Optimal mail certificates, introduced in [11], are efficient types of implicit certificates which offer many advantages over traditional (explicit) certificates. For example, an optimal mail certificate is small enough to fit on a two-dimensional digital postal mark together with a digital signature. This paper defines a general notion of security for implicit certificates, and proves that optimal mail certificates are secure under this definition.

Keywords: implicit certificate, authentication, postal payment, elliptic curve cryptography, provable security

1 Introduction

A certificate authority (CA) is a vital component of a secure public-key infrastructure. The primary role of the CA is to bind public keys to their legitimate owners by creating *certificates* for those users. Certificates are comprised of a data part which contains, at a minimum, a public key and a user identity, and the CA's signature on this data. If the CA's signature on the data part of a certificate is valid, then one has confidence that the data (and what the data implies) contained in the certificate is genuine; in particular one trusts that the user named in the certificate knows the private key associated with the public key.

Implicit certificates, introduced in the work of Günther [7] and Girault [6], are comprised of a user's identity I and some reconstruction public data P, which together with the CA's public key are used to reconstruct the user's public key. That is, the public key is not explicitly contained in the implicit certificate. The authenticity of a reconstructed public key is only established after it is subsequently used in a successful run of some protocol. In *identity-based* implicit certificate schemes, the user's private key is computed by the CA. In *self-certified* implicit certificate schemes, the user itself computes its private key and associated public key. For further details on this distinction, see [9]. This paper only considers self-certified implicit certificate schemes.

We consider the security aspects of the implicit certificate scheme introduced in [11], the *optimal mail certificate* scheme. This scheme has several desirable efficiency attributes, in particular, the bit length of an optimal mail certificate is short enough to fit in a two-dimensional digital postal mark together with an elliptic curve based digital signature with partial message recovery (a PVSSR signature [4, 8]). The major contribution of this paper is the formulation of a general definition for the security of implicit certificate schemes, and a proof that the optimal mail certificate scheme of [11] is secure under this definition.

The remainder of this paper is organized as follows. Our notation and some security issues of implicit certificates are covered in §2. §3 reviews the optimal mail certificate scheme. The security model and proof and presented in §4. Finally, some conclusions are drawn in §5.

2 Background

2.1 Notation

Our notation for the elliptic curve[1] domain parameters specifying the group, its order and its generator are as follows. Let q denote the order of the underlying finite field $\mathbb{F}_q$, and let E be an elliptic curve defined over $\mathbb{F}_q$. Let G denote a point in $E(\mathbb{F}_q)$, the *generator point*, and let n denote the order of G. We assume that n is prime. Thus $nG = \mathcal{O}$ and $G \neq \mathcal{O}$. The group $\langle G \rangle$ of points generated by G will be used in the optimal mail certificate scheme. We assume that the discrete logarithm problem in $\langle G \rangle$ is intractable. More precisely, this means that there is no probabilistic polynomial-time algorithm (polynomial in the security parameter $l = \lfloor \log_2 n \rfloor$) which on input $C \in_R \langle G \rangle$, $C \neq \mathcal{O}$, can output $c \in [1, n-1]$ satisfying $C = cG$ with non-negligible probability.

Let $c \in [1, n-1]$ be the CA's private key, and $C = cG$ the CA's public key. Similarly, $b \in [1, n-1]$ is Bob's private key, and $B = bG$ is Bob's public key. Let P denote Bob's reconstruction public data, which is an elliptic curve point. We refer to the party that reconstructs Bob's public key as Alice.

Let I denote some information that is included in Bob's certificate. In the case of an implicit certificate, I should typically include data such as

[1] Although the implicit certificate scheme studied in this paper can be described using any cyclic group such as $\mathbb{Z}_p^*$, the efficiency of elliptic curve groups are better matched with the efficiency of the optimal mail certificate scheme.

Bob's identifier, the CA's identifier, the validity period of the certificate, and possibly a serial number.

Let H denote a secure hash function, such as SHA-1. The selected hash function is an important part of the domain parameters of the system.

2.2 Implicit Certificates

Bob's implicit certificate is a pair (P, I) which, together with the CA's public key C, is used by Alice to reconstruct Bob's public key B. As in traditional, explicit certificates, Alice must trust the CA and the authenticity of C in order to arrive at the assurance that B is indeed Bob's public key. With explicit certificates, Alice verifies the signature with public key C, and thus is assured that B belongs Bob. However, such an explicit certificate alone is not sufficient for Bob to authenticate himself to Alice, because the explicit certificate is public information. To authenticate himself, Bob must demonstrate knowledge of his private key b through some secure cryptographic protocol such as a key agreement scheme or a digital signature scheme. The same holds true for an implicit certificate: to authenticate himself to Alice, Bob must demonstrate knowledge of his private key b. Once Bob accomplishes this, Alice is assured that B belongs to Bob, and also that Bob has authenticated himself in the protocol used to prove his knowledge of b. In other words, one distinction between explicit and implicit certificates is that, for the latter, the authentication to Alice of B belonging to Bob and the authentication to Alice that she is communicating with Bob are not separable.

3 Optimal Mail Certificates

In [11], Pintsov and Vanstone proposed an implicit certificate scheme which has significant efficiency advantages. We review this scheme, also depicted in Figure 1. Suppose that Bob wishes to obtain an implicit certificate from a CA. Bob selects $r \in_R [1, n-1]$, computes $R = rG$, and sends R to the CA over an authentic channel. The CA checks Bob's credentials according to its policies, and establishes certificate information I for Bob's implicit certificate. The CA then selects $k \in_R [1, n-1]$, and computes Bob's reconstruction public data $P = kG + R$. The CA then computes $h = H(P, I)$ and $s = hk + c \bmod n$.[2] Finally, the CA sends (P, I, s) to Bob. Bob sets his private key to be $b = hr + s \bmod n$ where

[2] This signing equation is a variant of the Schnorr signing equation [14].

again $h = H(P, I)$. Bob's public key, $B = bG$, can be reconstructed by Alice from the implicit certificate (P, I) using the equation

$$B = H(P, I)P + C,$$

which holds because $B = bG = (hr + s)G = (hr + hk + c)G = h(k + r)G + cG = hP + C$, where $h = H(P, I)$. Thus Bob knows the logarithm of $B = H(P, I)P + C$ with respect to the generator point G. Alice can reconstruct B using only the point P, the certificate information I, and the CA's public key C.

Figure 1. The optimal mail certificate scheme.

CA		Bob		Alice
		$r \in_R [1, n-1]$ $R = rG$		
	$\xleftarrow[\text{authentic}]{R}$			
$k \in_R [1, n-1]$ $P = R + kG$ $h = H(P, I)$ $s = kh + c \bmod n$				
	$\xrightarrow{(P,I,s)}$			
		$h = H(P, I)$ $b = rh + s \bmod n$		
			$\xrightarrow{(P,I)}$	
				$B = H(P, I)P + C$

Note that when Bob or any other party receives the response (P, I, s) to certificate request R, then it is possible to verify the authenticity of the response by checking the equation

$$H(P, I)R + sG = H(P, I)P + C.$$

Since this equation involves the CA's public key, an integer s, and a secure hash function, s would seem to depend on the private key of the CA in some way. Indeed this equation is similar to the equation used to verify digital signatures such as DSA and Schnorr signatures.

3.1 Applications

In environments where bandwidth is severely constrained, traditional X.509 certificates based on a digital signature algorithm such as RSA, DSA or ECDSA, may be too long to be transmitted in a public-key protocol. Instead, a shorter piece of information that identifies the certificate, such as the issuer and serial number or the eight bytes of the hash of the public key, is sent. However, this requires the recipient of the certificate to look up the certificate in some database, which would either be stored locally or remotely. Such a look-up may be costly, especially at remote locations or using limited resources. One method to alleviate this problem is random sampling, i.e., verifying only a small fraction of all certificates.

Optimal mail certificates are designed to be small enough to be sent completely, in order to avoid look-up and random sampling. Furthermore, their small size allows for larger local databases to be maintained, in case local look-up or random sampling is used.

The size of an optimal mail certificate (P, I) based on an elliptic curve over $\mathbb{F}_q$, where $q = 2^{163}$, is about 164 bits plus the length of the certificate information I. In comparison, the size of an explicit certificate containing a public key defined over the same elliptic curve and signed with ECDSA by a certificate authority with the public key on the same elliptic curve would be about 492 bits plus the length of the certification information. An explicit certificate based on RSA public keys and a digital signature using public keys of comparable security (namely, 1024 bits) would have length 2048 bits plus the length of the certificate information. For constrained environments, such as a two-dimensional bar code, even 492 bits may be too large for a reasonable density and size of bar code.

4 Security Model and Proof

In what follows we prove certain attributes of the optimal mail certificate scheme described in §3. Our proofs are in the random oracle model [3], in which the hash function is modeled by a perfectly random function. In reality, a hash function such as SHA-1 cannot be regarded as a random function. Nevertheless, a proof in the random oracle model suggests that if there is any attack against the scheme, then the attack must be specific to a particular hash function being used, or must process the description of the hash function itself. Intuitively, such attacks seem less feasible than generic attacks that simply invoke the hash function as a black-box. Indeed, almost all attacks in practice on such protocols are in fact generic attacks. For a more detailed discussion on the pros and cons of the random oracle model, see [2] and [5].

4.1 Security Model for Implicit Certificates

We formulate a notion of security for general implicit certificate schemes. The legitimate users in the system are denoted $\mathrm{Bob}_1, \mathrm{Bob}_2, \ldots$. The CAs in the system are denoted $\mathrm{CA}_1, \mathrm{CA}_2, \ldots$; CA_j's public key is C_j. We assume that all users have authentic copies of all CA public keys. User Bob_i's request for a certificate from CA_j is denoted (R_i, j). CA_j's response is denoted (P_i, I_i, s_i) which, together with C_j, is used by Bob_i to construct his public key B_i and associated private key b_i. Bob_i may make multiple requests for certificates from CA_j.

Definition 1. A (τ, ϵ)-adversary A *(of an implicit certificate scheme) is a probabilistic Turing machine which runs in time at most τ and interacts with the legitimate users and the CAs by performing each of the following operations any number of times:*

(i) receive a request (R_i, j) from Bob_i for an implicit certificate from CA_j; and

(ii) send a request $(R'_{i'}, j')$ to $CA_{j'}$, and receive response $(P'_{i'}, I'_{i'}, s'_{i'})$ from $CA_{j'}$.

With probability at least ϵ, A outputs a triple (P, I, b) such that b is the private key associated with the public key reconstructed from P, I and some C_k (that is, $bG = H(P,I)P + C_k$) such that either

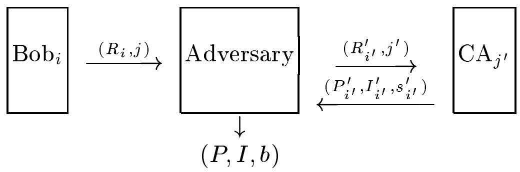
(i) (P, I) was never part of a response of CA_k; or

(ii) (P, I) was included in a response of CA_k to some request (R_i, j) originally from Bob_i.

A (τ, ϵ)-adversary is successful *if τ is polynomial in $l = \lfloor \log_2 n \rfloor$, and ϵ is a non-negligible function of l.*

The security model if depicted in Figure 2. All requests (R_i, j) from user Bob_i are sent through the adversary A, who may pass on the request unchanged to CA_j, or may choose to modify the request point R_i, or the identity i of the requester, or the designated CA. In this model, the adversary A is deemed successful if it generates an implicit certificate that is a forgery in either one of the following two senses. In the first sense, A generates an implicit certificate and an associated private key that was not issued by a CA. This may be regarded as a successful attack by A against that CA. In the second sense, A generates an implicit certificate and a private key that was issued by a CA for a request of some user Bob_i. This may be regarded as a successful attack by A against Bob_i.

Figure 2. A security model for implicit certificates.



Note that the security model is stronger in some regards that what may be required in practice. For example, Definition 1 does not insist that the string I output by A satisfy some formatting specification as might be the case in practice. Also, the model does not insist that communications from the requester to the CA be authentic.

Definition 2. *An implicit certificate scheme is* secure *if there does not exist a successful adversary of the kind described in Definition 1.*

A more concrete definition would be that an implicit certificate scheme is (τ, ϵ)-*secure* if there does not exist such an adversary A that runs in time at most τ and succeeds with probability at least ϵ. However, in this paper we will only consider the asymptotic form of security as given in Definition 2 above.

4.2 Security Proof

In order to prove the security of optimal mail certificates, we shall work in the random oracle model. Several signature schemes have been proved secure in the random oracle model. We shall use the following security result for Schnorr signature schemes.

Theorem 1 (Pointcheval and Stern [12]). *In the random oracle model, the Schnorr family of signature schemes over a group $\langle G \rangle$ is secure if the discrete logarithm problem in $\langle G \rangle$ is intractable.*

The proof of Theorem 1 follows easily from the proofs given in [12]. The proof involves the "forking lemma" and a reduction from a signature-forging algorithm to a discrete logarithm-solving algorithm. We shall use such a reduction in our proof. Theorem 1 is true for a variety of signing equations that can be used in the family of Schnorr signature schemes, and in particular it is true for the signing equation used in optimal mail certificates. That is, the following variant of the Schnorr signature is secure: For a message m and random oracle H, a signature (R, s) is generated by Bob with $R = kG$ for random k and $s = H(R, m)k + b \bmod n$ where b is Bob's private key. To verify the signature, one checks that $sG = H(R, m)R + B$, where B is Bob's public key.

Theorem 2. *In the random oracle model, the optimal mail certificate scheme is a secure implicit certificate scheme provided that the discrete logarithm problem in $\langle G \rangle$ is intractable.*

Proof. Assume that the optimal mail certificate scheme is not secure in the case where the hash function H is a random oracle. Then there exists a successful (τ, ϵ)-adversary A. We construct a polynomial-time algorithm S that uses A as a subroutine to compute logarithms in $\langle G \rangle$ with non-negligible probability.

The input to S consists of a discrete logarithm challenge $C \in_R \langle G \rangle$, $C \neq \mathcal{O}$, and the desired output of S is an integer $c \in [1, n-1]$ such that $C = cG$. We shall construct S in two stages. The first stage S_1 takes as input (C, m, H_1) where m is a random message, and H_1 is a random oracle independent of H. S_1 can use A as a subroutine. The desired output of S_1 is either (i) an integer $c \in [1, n-1]$ such that $C = cG$, or (ii) an ordered pair (P, b) such that $bG = H_1(P, m)P + C$ (i.e., (P, b) is a signature of message m with respect to the public key C). If case (i) occurs, then S outputs c and terminates. If case (ii) occurs, then Theorem 1 is used to reduce the signature forger S_1 to a discrete logarithm solver in order to extract c. If this stage is successful, then S outputs c and terminates.

To find c, algorithm S_1 runs algorithm A. Algorithm A expects there to be one or more CAs, each with a public key for which A is not given the private key, and zero or more requester Bob$_i$ making one or more requests R_i for which A is not given the discrete logarithm r_i. Algorithm S_1 randomly selects one of the CA public keys or one of the requests to be the challenge point C which is the input of S. The other request points and CA public keys can be selected by S_1 according to the normal procedure of selecting a random secret integer and multiplying G by this value. Let t be the total number of CA public keys and requests. We shall see that there will be a ϵ/t probability that A can be used to obtain c or a forgery of a signature with public key C.

Since A can request a certificate from the CA with public key C (if S_1 has selected such a CA) and expect a legitimate response, S_1 must supply a response that seems legitimate at least from A's perspective.

(Otherwise A is not guaranteed success, and S_1 may not find A useful to find c.) However, S_1 does not know the private key c associated with C. But since H is a random oracle, S_1 can simulate the role of the CA and answer A's certificate requests without knowing c by careful pre-selection of the random values of H. Algorithm S_1 simulates the role of the CA as follows: given a request R_i for a certificate with certificate information I_i, S_1 generates integers $s_i, h_i \in_R [0, n-1]$ and computes $P_i = R_i + h_i^{-1}(s_i G - C)$, where the inverse is computed modulo n. S_1 defines $H(P_i, I_i) = h_i$ and returns the triple (P_i, I_i, s_i) as the response to A's request. Since $H(P_i, I_i)R_i + s_i G = H(P_i, I_i)P_i + C$ holds, the response to the certificate request appears legitimate from A's perspective. Furthermore, the hash function will be random from A's perspective because the value h_i was initially chosen randomly.

The adversary A is of course allowed to query H directly. Given a hash query input, say, (P_A, I_A), which has not been previously queried or determined as above, S_1 outputs $H_1(P_A, m)$ where m is the message on which it is trying to forge a signature. Clearly, the distribution of the simulated hash values generated by S will be indistinguishable to A from the distribution of hash values generated by a random oracle.

Suppose that A is successful. Then A returns a triple (P, I, b) such that $bG = hP + C_k$ for some k, with $h = H(P, I)$, such that either:

(i) (P, I) is a certificate created by CA_k for a request from Bob_i; or
(ii) (P, I) is a certificate which was not issued by CA_k.

Assume we are in the first case. Then there is at least a $1/t$ probability that the request R_i of Bob_i was the challenge point C given as input to the algorithm S_1. The private key b of Bob_i discovered by A satisfies $b = rh + s \bmod n$. But $c = r$, and S_1 can observe s as CA_k's response. Thus S_1 can compute $c = h^{-1}(b - s) \bmod n$.

Assume we are in the second case. Then there is at least a $1/t$ probability that public key C_k of CA_k is the challenge point C given as input to the algorithm S_1. We can assume that (P, I) was an input query to the random oracle hash H, because otherwise the equation $H(P, I)P + C = bG$ will hold with negligible probability, contradicting the assumption that ϵ is non-negligible. Thus $H(P, I) = H_1(P, m)$ by definition of the simulation. But now (P, b) is a signature of the message m.

There is a minor problem if during execution of S_1 with A, the message (P, I) appears first as a direct query to H, and subsequently as a certificate constructed during the simulation of a CA. Since the values s and r are chosen randomly during simulation of the CA, this event happens with negligible probability. Nevertheless, in this case S_1 can simply start over.

Clearly, if A runs in polynomial time and succeeds with non-negligible probability then so will S_1. By Theorem 1 and above, if A runs in polynomial time and succeeds with non-negligible probability then so will S. But by hypothesis, it was assumed that no such S for solving discrete logarithms in $\langle G \rangle$ existed. Therefore no adversary A exists in the random oracle model unless discrete logarithms in $\langle G \rangle$ can be efficiently solved.

$\square$

4.3 Secure Use of Optimal Mail Certificates

In §4.2 we argued that it is unlikely that an adversary can produce a "new" (P, I) pair such that the adversary knows the associated private key. Since public-key protocols are designed to be secure unless the adversary knows the private key associated with a user's public key, one might conclude that it is safe to now use optimal mail certificates instead of traditional certificates. However, it is possible that the use of optimal mail certificates in a given protocol could lead to insecurities. For example, if optimal certificates are used in conjunction with the Nyberg-Rueppel signature scheme [10], it would certainly be undesirable if an adversary could produce a (P, I) pair, and a triple (r, s, M) such that the pair (r, s) is a valid signature on the message M by the user with public key $H(P, I)P + C$. Nothing we have said thus far precludes this possibility. Similar questions could be posed concerning the use of optimal mail certificates in other public-key protocols such as key agreement and public-key encryption.

4.4 Denial-of-Service Attacks

One difference between traditional explicit certificates over implicit certificates is that when presented with a valid explicit certificate, we know that the certificate belongs to *someone*. Specifically, up to forgery of the signature scheme used to generate certificates, a valid certificate containing identifier I is proof that the CA signed this certificate for I, and so the party identified by I knows the private key associated with the public key included in the certificate. This guarantee does not hold with implicit certificates, such as the ones described in §3. In fact, an arbitrary pair (P, I) can be given to Alice to reconstruct a public key $B = H(P, I)P + C$. But, it should be infeasible to find such a pair (P, I) where the discrete logarithm b with respect to the generator point G of the reconstructed public key B is known. Since authentication of Bob is only completed once Bob has demonstrated knowledge of b in a subsequent protocol run, such arbitrary "faked" (P, I) will not deceive Alice.

The above discussion suggests a denial-of-service type attack, where Alice is flooded with protocol requests using "faked" implicit certificates. The fact that the logarithm of the faked public key is unknown is revealed only after Alice has performed most of the protocol. Of course, a similar attack can be launched in a system using regular certificates. In this case, though, the attacker floods Alice with various certificates belonging to other entities. Alice verifies the digital signature on each certificate received. The certificates are valid, but the attacker does not know the private keys of the associated public keys. In either case, Alice is forced to perform many cryptographic operations with no useful result, which may prevent her from doing something useful like authenticating a legitimate user.

5 Conclusions

Optimal mail certificates are more efficient in some respects than traditional explicit certificates and other types of implicit certificates. We have proposed a security definition for implicit certificate schemes, and proved that optimal mail certificates are secure under this definition. What remains to be proven is that the use of optimal mail certificates in conjunction with specific public-key protocols such as signature schemes does not in any way reduce the security of the protocols.

Acknowledgements

The authors would like to thank Alfred Menezes and Doug Stinson for their careful comments on earlier drafts of this paper.

References

1. B. Arazi, "An identity-based DL/EC key agreement technique", Presentation to IEEE P1363a, available from http://grouper.ieee.org/groups/1363/addendum.html
2. M. Bellare, " Practice-oriented provable-security", *Proceedings of First International Workshop on Information Security (ISW 97)*, 1998.
3. M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols", *1st ACM Conference on Computer and Communications Security*, 1993 pp. 62-73.

4. D. Brown and D. Johnson, "Formal security proofs for a signature scheme with partial message recovery", preprint, 2000.
5. R. Canetti, O. Goldreich and S. Halevi, "The Random Oracle Methodology, Revisited (Preliminary Version)", *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing*, 1998, pp. 209-218.
6. M. Girault, "Self-certified public keys", *Advances in Cryptology – Eurocrypt'91*, 1991, pp. 490-497.
7. C.G. Günther, "An identity-based key-exchange protocol", *Advances in Cryptology – Eurocrypt'89*, 1989, pp. 29-37.
8. IEEE P1363a, "Standard Specifications for Public-Key Cryptography: Additional Techniques", Working Draft, August 16, 2000.
9. A. Menezes, P. van Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.
10. K. Nyberg and R. Rueppel, "Message recovery for signature schemes based on the discrete logarithm problem", *Designs, Codes and Cryptography*, **7** (1996), pp. 61-81.
11. L. Pintsov and S. Vanstone, "Postal Revenue Collection in the Digital Age", *Proceedings of Financial Cryptography 2000*, to appear.
12. D. Pointcheval and J. Stern, "Security proofs for signature schemes", *Advances in Cryptology – Eurocrypt'96*, 1996, pp. 387-398.
13. D. Pointcheval, J. Stern, "Security arguments proofs for Digital Signatures and Blind Signatures", *Journal of Cryptology*, 2000, pp. 361-396.
14. C. Schnorr, "Efficient signature generation by smart cards", *Journal of Cryptology*, **4** (1991), pp. 161-174.