Invisibility and Anonymity of Undeniable and Confirmer Signatures

Steven D. Galbraith*[1] and Wenbo Mao[2]**

[1] Mathematics Department, Royal Holloway University of London,
Egham, Surrey TW20 0EX, UK.
Steven.Galbraith@rhul.ac.uk

[2] Mathematics, Cryptography and Security Group
Hewlett-Packard Laboratories, Bristol
Filton Road, Stoke Gifford, Bristol BS34 8QZ, UK.
wm@hplb.hpl.hp.com

Abstract. Traditionally, the strongest notion of security for undeniable and confirmer signatures is invisibility under adaptive attacks. This security property was promoted by Camenisch and Michels and they provided schemes with this property. Gennaro, Krawczyk and Rabin (GKR) developed an RSA-based scheme which is much more efficient than the schemes of Camenisch and Michels, but it does not have invisibility. We give an RSA-based scheme which is as efficient as the GKR scheme, and which has invisibility.

We suggest that anonymity is the most relevant security property for undeniable and confirmer signatures. We give a precise definition of anonymity for undeniable and confirmer signatures in the multi-user setting and show that anonymity and invisibility are closely related. Finally, we show that anonymity can be achieved even when the parties use completely different cryptographic primitives.

Keywords: Undeniable signatures, confirmer signatures, RSA, invisibility, anonymity.

1 Introduction

Undeniable signatures [9, 10] are public key digital signatures which cannot be verified without interacting with the signer. Confirmer signatures [12] are undeniable signatures where signatures may also be verified by interacting with an entity called the confirmer who has been designated by the signer. Implicit in these notions is the principle that validity of a signature cannot be determined without some interaction. Undeniable and confirmer signatures have been used in various applications, including auctions [21, 22].

The strongest notion of security in the literature for undeniable and confirmer signatures is that of 'invisibility', which was introduced by Chaum, van Heijst and Pfitzmann [11]. This is essentially the inability to determine whether a given message-signature pair is valid for a given user. In [11] invisibility is defined in terms of simulatability. In [7] this notion is phrased in terms of distinguishing whether a signature s corresponds to a message m_0 or m_1.

The history of undeniable and confirmer signatures is no different from the history of other public key techniques: the strong notions of security were introduced after the early schemes had been invented, and the original schemes did not provide these security properties. Camenisch and Michels [7] gave a general method to obtain a secure confirmer signature scheme out of any sufficiently secure signature scheme and any sufficiently secure encryption scheme. Their methods can be used to obtain secure schemes based on RSA or discrete logarithms. The drawback of [7] is that efficiency is sacrificed to obtain security.

* This author thanks Hewlett-Packard Laboratories, Bristol and the EPSRC for support.

** This author's research is partially funded by the EU Fifth Framework Project IST-2001-324467 "CASENET".

Much more efficient RSA-based undeniable and confirmer signatures were developed by Gennaro, Krawczyk and Rabin [15] and Galbraith, Mao and Paterson [14]. However, these schemes were not developed to provide invisibility. One of the main aims of this paper is to present a new RSA-based scheme which has the invisibility security property (i.e., is as secure as the scheme of [7]) and yet which is essentially as efficient as the schemes of [14, 15]. Hence, our work is analogous to [3] which gave the first signature schemes which are as efficient as RSA/Rabin and secure in a very strong sense.

We believe that anonymity rather than invisibility should be considered as the main security property for undeniable and confirmer signatures in the multi-user setting. Informally, this security property is as follows. Imagine a system with n users and suppose an adversary is given a valid message-signature pair and is asked to determine which user generated the signature. By running signature confirmation or denial protocols with a given user (or their designated confirmer) one can determine whether or not the user generated the signature. An undeniable or confirmer signature scheme has the anonymity property if it is infeasible to determine whether a user is or is not the signer of the message without interacting with that user or with the $n-1$ other users. A more precise definition of anonymity is given in Definitions 3 and 4 where the problem is distilled down to the case of two users. We show that anonymity and invisibility are equivalent notions for schemes with certain additional properties.

The standard solution to the problem of anonymous signatures is group signatures. In this case a set of users is identified as a 'group' and a certain trusted party (the group manager) issues parameters for the scheme. A significant drawback of group signatures, for some applications, is the issue (highlighted by Ateniese and Tsudik [1]) of adding and removing members from a group. While there are recent papers on this problem, all of them have drawbacks (the schemes of Ateniese, Song and Tsudik [2] and Bresson and Stern [5] require the signature size to grow linearly in the number of removed members, while the ideas of Camenisch and Lysyanskaya [8] require the group manager to send every group member a new signing key when a member is removed).

We believe that confirmer signatures which have the anonymity property are preferable to group signatures in certain applications, as revocation can be handled by the PKI in the usual way. In both cases there is a trusted third party (confirmer or group manager) who can handle disputes.

General techniques show that anonymity can be achieved (for example, by combining the results of Okamoto [19] and Bellare, Boldyreva, Desai and Pointcheval [4]). However, our schemes are significantly more efficient than schemes which would arise using these general techniques.

For confirmer signature schemes we stress that we do not study the issue of whether the signature reveals who the designated confirmer is (though our solutions do provide anonymity for the confirmer).

1.1 Plan of the Paper

In Section 2 we clarify what we mean by undeniable and confirmer signature schemes. In Section 3 we generalise the notion of invisibility to one which we believe is more natural in the multi-user case. In Section 4 we give a precise definition for anonymity and prove two of our main results: that anonymity and generalised invisibility are essentially equivalent.

Sections 5, 6 and 7 form the technical heart of the paper. We give two attacks on the anonymity of the RSA-based undeniable and confirmer signature schemes of Gennaro, Krawczyk and Rabin [15] and Galbraith, Mao and Paterson [14]. We give a new RSA-based undeniable/confirmer signature scheme. Our main result is Theorem 5 which shows that our scheme has the anonymity property. In Section 8 we sketch some other properties of our new scheme.

In Section 9 we discuss the invisibility and anonymity of some other schemes in the literature. In Sections 10 and 11 we give a version of the Chaum and van Antwerpen scheme which has invisibility and anonymity.

In Section 12, we discuss anonymity in the extremely general situation where participants may use completely different undeniable and confirmer signature schemes. We argue that anonymity can be obtained even in this setting.

2 Undeniable and Confirmer Signature Schemes

An *undeniable signature scheme* consists of two algorithms, namely Gen and Sign, and two protocols, namely Confirm and Deny. For every choice of the security parameter k there is a public-key space $\mathcal{K}$, a message space $\mathcal{M}$ and a signature space $\mathcal{S}$. For our applications we stress that the space $\mathcal{S}$ must depend only on the security parameter k and not on a specific public key.

Gen is a randomised algorithm which takes as input a security parameter k and outputs a public key $pk \in \mathcal{K}$ and a corresponding secret key sk.

Sign is an algorithm (in our case it must be randomised) which takes as input a secret key sk and a message $m \in \mathcal{M}$ and outputs a signature $s = \mathrm{Sign}_{sk}(m) \in \mathcal{S}$.
In general there are many valid signatures for any pair $(pk, m) \in \mathcal{K} \times \mathcal{M}$.

Confirm is a protocol between a signer and a verifier which takes as input a message $m \in \mathcal{M}$, a signature $s \in \mathcal{S}$ and a (certified) public key pk and allows the signer to prove to a verifier that the signature s is valid for the message m and the key pk. If the verifier has a suitable public key then the proof may be taken to be a non-interactive, designated-verifier proof [16].

Deny is a protocol which takes $m \in \mathcal{M}$, $s \in \mathcal{S}$ and $pk \in \mathcal{K}$ and allows a signer to prove to a verifier that the given signature is not valid for that key. In some schemes the denial protocol is the same as the confirmation protocol.

A *confirmer signature scheme* is essentially the same as above, except the role of confirmation and denial can also be performed by a third party called a 'confirmer'. The significant modification is that the algorithm Gen now produces a confirmation key ck which is needed for the Confirm and Deny protocols.

The literature on confirmer signatures is inconsistent on whether the original signer has the ability to confirm and/or deny signatures. Camenisch and Michels [7] claim that it is undesirable for signers to be able to confirm or deny their signatures. We have a contrary opinion, that it is important for signers to be able to confirm and/or deny signatures (e.g., to clear their name by denying signatures that are not genuine). The schemes of [7, 12, 18] do not allow users to deny signatures, whereas the schemes of [9, 10, 14, 15] do allow this. In any case, these distinctions have no bearing on the discussion of the invisibility and anonymity of the schemes.

The following properties will be used later. Property A is essentially that, for a fixed key and varying messages, signatures look random. Property B is essentially that, for a fixed message and varying keys, signatures look random.

Property A: Let k be any value of the security parameter. Let (pk, sk) be any output of the Gen algorithm for the security parameter k. Consider the uniform distribution on $\mathcal{M}$. Then the distribution on $\mathcal{S}$ corresponding to the random variable $\mathrm{Sign}_{sk}(m)$ is indistinguishable from uniform.

Property B: Let k be any value of the security parameter and let $m \in \mathcal{M}$ be an arbitrary message. Consider the distribution on $\mathcal{K}$ induced by the randomised algorithm Gen. Then for pk chosen at random from $\mathcal{K}$ according to this distribution (with the corresponding secret key sk), the distribution on $\mathcal{S}$ corresponding to the random variable $\mathrm{Sign}_{sk}(m)$ is indistinguishable from uniform.

A typical example in our situation of a distribution which is indistinguishable from uniform is the following. Let $\mathcal{S}$ be the set of all binary strings of length $2k$, let N be a k-bit RSA modulus, and consider the uniform distribution on the set $\mathcal{S}' = \{s \in \mathcal{S} : \gcd(s, N) = 1\}$ (where a binary string is interpreted as an integer in the natural way). Then, given N and an algorithm which outputs polynomially many randomly sampled elements from $\mathcal{S}$ or $\mathcal{S}'$, it is infeasible to determine which of the two sets is being sampled.

3 Generalised Invisibility

We give a definition of invisibility which is more suitable for our purposes. There is one subtlety in this definition: For the schemes we will consider, signatures are padded to a fixed bitlength in a way which is malleable by an adversary. For example, with our RSA-based scheme, a signature is naturally a number modulo N (where N is part of a user's public key) and this number is extended to a longer bitstring by adding a multiple of N. Now, this process is malleable in the sense that an adversary can add or subtract N from a given bitstring to obtain another, equally valid, signature. This leads to a trivial adaptive attack on the scheme.

More precisely, signatures lie in equivalence classes whose structure is known to all users, and which depend on the particular public key. Hence we cannot allow an adaptive adversary to initiate confirm or denial protocols on any representative of the equivalence class of the challenge signature. Fortunately, these equivalence relations are always easily computed and so one can recognise if an adversary tries to make a query of this form.

Definition 1. *Let* (Gen,Sign,Confirm,Deny) *be an undeniable or confirmer signature scheme. An adversary* $\mathbf{D}$ *is said to be an* **invisibility distinguisher** *under an adaptive chosen message attack if it behaves as follows.*

Fix a value of the security paramter k (this determines the signature space $\mathcal{S}$). Let $(pk, sk) \leftarrow \mathrm{Gen}(1^k)$ be a key pair. The input to $\mathbf{D}$ is pk. The distinguisher $\mathbf{D}$ is permitted to interact with the hash function oracle(s), to obtain signatures on messages of its choice and to run signature verification and denial protocols (with the signer or a confirmer as appropriate) on elements of $\mathcal{S}$ of its choice. At some point $\mathbf{D}$ constructs a message m and requests a challenge s. The challenge s depends on the outcome of a hidden coin toss. If the hidden bit b is 0 then $s = \mathrm{Sign}_{sk}(m)$ and if the hidden bit is 1 then s is chosen uniformly at random from the signature space $\mathcal{S}$. The interaction with the cryptosystem continues with the exception that verification and denial protocols cannot be executed on any pair (m, s') in the equivalence class of the challenge message-signature pair (m, s). The output of $\mathbf{D}$ is a guess b' for the hidden bit b.

A distinguisher $\mathbf{D}$ with output b' is said to have **advantage** $\mathrm{Adv}(\mathbf{D}) = \epsilon(k)$, *if with probability at least $1/2 + \epsilon(k)$, we have $b' = b$.*

Definition 2. *An undeniable or confirmer signature scheme has* **invisibility** *if there is no polynomial time distinguisher $\mathbf{D}$ as in Definition 1 which has non-negligible advantage (i.e., given any polynomial $p(k)$ we have $\epsilon(k) < 1/p(k)$ for sufficiently large k).*

We now show that generalised invisibility is equivalent to the definition of invisibility given in [7] for certain schemes. First we recall the definition of invisibility given in [7]: the distinguisher is almost identical to Definition 1 above except that it presents two messages m_0, m_1 and the challenge is $s = \mathrm{Sign}_{sk}(m_b)$ where b is the hidden bit.

Note that these results hold in the standard model of computation and that the security reductions require essentially no added computation.

Theorem 1. *Let $\mathcal{U}$ be an undeniable or confirmer signature scheme. If $\mathcal{U}$ has invisibility in the sense of Definition 2, then $\mathcal{U}$ has invisibility in the sense of [7].*

Proof. Let $\mathcal{A}$ be a distinguisher in the sense of [7]. We build a distinguisher $\mathbf{D}$ using $\mathcal{A}$.

The input to $\mathbf{D}$ is pk and we execute $\mathcal{A}$ on this value. Queries made by $\mathcal{A}$ are passed on by $\mathbf{D}$. Eventually $\mathcal{A}$ outputs a pair (m_0, m_1). At this point flip a coin to obtain a bit b' and take $m_{b'}$ as the output of $\mathbf{D}$. The challenge s received by $\mathbf{D}$ is $\mathrm{Sign}_{sk}(m_{b'})$ in the case $b = 0$ or is a random element of $\mathcal{S}$ in the case $b = 1$. Pass s to $\mathcal{A}$ as the challenge. Continue to answer queries by $\mathcal{A}$ as before.

Finally, $\mathcal{A}$ outputs its guess b''. If $b'' = b'$ then output $b''' = 0$ by $\mathbf{D}$, and if $b'' \neq b'$ then output $b''' = 1$.

We have

$$\begin{aligned}\mathrm{Adv}(\mathbf{D}) &= \Pr(b''' = b) - \frac{1}{2}\\ &= \Pr(b''' = 0|b = 0)\Pr(b = 0) + \Pr(b''' = 1|b = 1)\Pr(b = 1) - \frac{1}{2}\\ &= \Pr(b'' = b'|b = 0)\frac{1}{2} + \Pr(b'' \neq b'|b = 1)\frac{1}{2} - \frac{1}{2}.\end{aligned}$$

Now, when $b = 0$ then the game is indistinguishable from a real attack, and so $\Pr(b'' = b'|b = 0) = \frac{1}{2} + \mathrm{Adv}(\mathcal{A})$.

On the other hand, when $b = 1$ then, with overwhelming probability, s is not a valid signature for either m_0 or m_1. It follows that the hidden bit b' is independent of the view of $\mathcal{A}$. Hence, $\Pr(b'' \neq b'|b = 1) \approx \frac{1}{2}$ where the notation $\approx$ denotes equality up to negligible terms[1] Therefore

$$\mathrm{Adv}(\mathbf{D}) \approx \left(\frac{1}{2} + \mathrm{Adv}(\mathcal{A})\right)\frac{1}{2} + \frac{1}{2}\frac{1}{2} - \frac{1}{2} = \frac{1}{2}\mathrm{Adv}(\mathcal{A})$$

and the result follows. □

Theorem 2. *Let $\mathcal{U}$ be an undeniable or confirmer signature scheme which satisfies Property A above. If $\mathcal{U}$ has invisibility in the sense of [7], then $\mathcal{U}$ has invisibility in the sense of Definition 2.*

Proof. We first assume that the distribution on $\mathcal{S}$ mentioned in Property A is exactly uniform. Let $\mathbf{D}$ be an distinguisher as in Definition 1. We must transform $\mathbf{D}$ into a distinguisher $\mathcal{A}$.

The input to $\mathcal{A}$ is the key pk and so we execute $\mathbf{D}$ on this key. Queries made by $\mathbf{D}$ are passed on as queries made by $\mathcal{A}$. When $\mathbf{D}$ produces a message m then select another message $m' \in \mathcal{M}$ uniformly at random and use m, m' (in either order) as the output of $\mathcal{A}$. The challenge received is s which is passed back to $\mathbf{D}$. Note that, s is a signature on either m or m', and by our assumption, the distribution of $s \in \mathcal{S}$ in the latter case is uniform.

The simulation then proceeds in the obvious way and is indistinguishable from a real game. We clearly have $\mathrm{Adv}(\mathbf{D}) = \mathrm{Adv}(\mathcal{A})$.

Now, we consider the case when the distribution is merely indistinguishable from uniform. Suppose we are given an adversary $\mathbf{D}$ as in Definition 1 and suppose that the advantage of $\mathbf{D}$ in the game played in the first half of the proof is non-negligibly different from the advantage of $\mathbf{D}$ in a real attack. Then it is straightforward to convert $\mathbf{D}$ into an algorithm which distinguishes the two distributions on $\mathcal{S}$. By Property A we conclude that $\mathrm{Adv}(\mathbf{D}) \approx \mathrm{Adv}(\mathcal{A})$. □

4 Anonymity

We give a rigorous definition for anonymity. The first step is to distill the problem down to the case of just two users (a scheme with the anonymity property for two users can easily be shown to be secure in the case of n users[2]). Note that if a signature is known to be valid for some user then the identity of the signer can be obtained by executing a signature confirmation protocol with that user, or by executing a signature denial protocol with the other user.

---

[1] The negligible quantity is the probability, over random $pk \in \mathcal{K}, \{m_0, m_1\} \subseteq \mathcal{M}$ and $s \in \mathcal{S}$, that s is a valid signature on at least one of m_i for pk. This probability depends on the scheme under consideration and so it is impossible to be more precise here. This probability must be negligible for any scheme which is secure against forgery of signatures.

[2] For general results n must be bounded by a polynomial in the security parameter k, but for some particular schemes a more tight security reduction is easily obtained.

Definition 3. *Let* (Gen,Sign,Confirm,Deny) *be an undeniable or confirmer signature scheme. An adversary* $\mathbf{D}$ *is said to be an* **anonymity distinguisher** *under an adaptive chosen message attack if it behaves as follows.*

Fix a value of the security parameter k *(this fixes the signature space* $\mathcal{S}$*). Let* $(pk_0, sk_0) \leftarrow \mathrm{Gen}(1^k)$ *and* $(pk_1, sk_1) \leftarrow \mathrm{Gen}(1^k)$ *be two key pairs. The input to* $\mathbf{D}$ *is the pair* (pk_0, pk_1)*. The distinguisher* $\mathbf{D}$ *is permitted to interact with the hash function oracle(s), to obtain signatures on messages of its choice, and to run signature verification and denial protocols (with the signer or a confirmer as appropriate) with respect to both of these public keys on elements of* $\mathcal{S}$ *of its choice. At some point* $\mathbf{D}$ *constructs a message* m *and requests a challenge signature* $s \leftarrow \mathrm{Sign}_{sk_b}(m)$ *where the bit* $b \in \{0,1\}$ *is hidden from* $\mathbf{D}$*. The interaction with the cryptosystem continues with the exception that verification and denial protocols cannot be executed with respect to key* pk_i *on any pair* (m, s') *in the equivalence class corresponding to key* pk_i *of the challenge message-signature pair* (m, s)*. The output of* $\mathbf{D}$ *is a guess* b' *for the hidden bit* b*.*

A distinguisher $\mathbf{D}$ *with output* b' *is said to have* **advantage** $\mathrm{Adv}(\mathbf{D}) = \epsilon(k)$*, if with probability at least* $1/2 + \epsilon(k)$*, we have* $b' = b$*.*

Definition 4. *An undeniable or confirmer signature scheme has* **anonymity** *if there is no polynomial time distinguisher* $\mathbf{D}$ *as in Definition 3 which has non-negligible advantage (i.e., given any polynomial* $p(k)$ *we have* $\epsilon(k) < 1/p(k)$ *for sufficiently large* k*).*

We now show that anonymity and invisibility are essentially equivalent. Note that both Theorems below are proved in the standard model of computation and that the security reductions require essentially no added computation.

Theorem 3. *Let* $\mathcal{U}$ *be an undeniable or confirmer signature scheme which has Property B. If* $\mathcal{U}$ *has anonymity, then* $\mathcal{U}$ *has invisibility.*

Proof. First assume that the distribution in Property B is precisely uniform. Let $\mathbf{D}_I$ be an invisibility adversary, we will create an anonymity adversary $\mathbf{D}_A$ from $\mathbf{D}_I$. The input to $\mathbf{D}_A$ is (pk_0, pk_1) and we run $\mathbf{D}_I$ on pk_0. The queries made by $\mathbf{D}_I$ can all be passed on as $\mathbf{D}_A$ queries. Note that pk_1 is independent of the view of $\mathbf{D}_I$.

Eventually $\mathbf{D}_I$ produces a message m and requests a challenge. The message m is output by $\mathbf{D}_A$ and the challenge s is received. Recall that s is $\mathrm{Sign}_{sk_0}(m)$ in the case $b = 0$ and is $\mathrm{Sign}_{sk_1}(m)$ in the case $b = 1$. In the case $b = 1$, since pk_1 is uniformly chosen at random from $\mathcal{K}$, our assumption implies that s is a uniformly random element of $\mathcal{S}$. In other words, the value s is compatible with the game $\mathbf{D}_I$ is designed to play.

Further queries by $\mathbf{D}_I$ are passed on by $\mathbf{D}_A$. Finally, $\mathbf{D}_I$ outputs a guess b' for b. This bit is used by $\mathbf{D}_A$ as its guess for the hidden bit b. It is clear that

$$\mathrm{Adv}(\mathbf{D}_A) = \mathrm{Adv}(\mathbf{D}_I).$$

Now consider the case where the distribution in Property B is indistinguishable from uniform. Let $\mathbf{D}_I$ be an invisibility distinguisher in the sense of Definition 1. If the advantage of $\mathbf{D}_I$ in the game described earlier in the proof is non-negligibly different from the advantage of $\mathbf{D}_I$ in a real attack then $\mathbf{D}_I$ can be easily transformed into a distinguisher for Property B. It follows that

$$\mathrm{Adv}(\mathbf{D}_A) \approx \mathrm{Adv}(\mathbf{D}_I).$$

which completes the proof. □

Theorem 4. *Let* $\mathcal{U}$ *be an undeniable or confirmer signature scheme which has invisibility, then* $\mathcal{U}$ *has anonymity.*

Proof. Let $\mathbf{D}_A$ be an anonymity distinguisher. We will construct an invisibility distinguisher $\mathbf{D}_I$ from $\mathbf{D}_A$. The input to $\mathbf{D}_I$ is a key pk_0. We execute Gen to construct another public key pk_1 (note that the secret key sk is known to $\mathbf{D}_I$).

Flip a coin to obtain a bit b'. If $b' = 0$ then run $\mathbf{D}_A$ on the pair (pk_0, pk_1) and if $b' = 1$ then run $\mathbf{D}_A$ on the pair (pk_1, pk_0).

Queries made by $\mathbf{D}_A$ are answered in the obvious way: queries with respect to pk_0 are passed on by $\mathbf{D}_I$, and queries with respect to pk_1 are handled using knowledge of the secret key.

Eventually $\mathbf{D}_A$ produces a message $m \in \mathcal{M}$ and requests a challenge. We use m as the challenge for $\mathbf{D}_I$ and receive a value $s \in \mathcal{S}$ which is $\mathrm{Sign}_{pk_0}(m)$ in the case $b = 0$ or is a uniformly random element of $\mathcal{S}$ in the case $b = 1$. The challenge s is passed to $\mathbf{D}_A$. In the case $b = 0$ we have that s is a valid signature on m for pk_0 and with overwhelming probability is not a valid signature on m for pk_1. In the case $b = 1$ we have that s is a random element of $\mathcal{S}$ and so, with overwhelming probability, s is not a valid signature on m for either public key. The queries by $\mathbf{D}_A$ continue to be handled as before, except we must be careful not to allow confirm or deny protocols to be run on elements of the equivalence class of the challenge signature with respect to key pk_1.

Finally, $\mathbf{D}_A$ outputs a guess b''. If $b'' = b'$ then output 0 as the guess for the hidden bit b, and if $b'' \neq b'$ then output 1.

It remains to compute the advantage of $\mathbf{D}_I$. First note that in the case $b = 0$

$$\mathrm{Adv}(\mathbf{D}_A) = \Pr(b'' = b') = \frac{1}{2} + \epsilon.$$

Now, the advantage of $\mathbf{D}_I$ is

$$\mathrm{Adv}(\mathbf{D}_I) = \Pr(b'' = b' | b = 0) \Pr(b = 0) + \Pr(b'' \neq b' | b = 1) \Pr(b = 1)$$

and $\Pr(b'' \neq b' | b = 1) \approx \frac{1}{2}$ (where, again, $\approx$ means 'equal up to negligible factors' since there is the negligible chance that the random s is a valid signature on m) since, in the case $b = 1$, the hidden bit b' is independent of s. It follows that

$$\mathrm{Adv}(\mathbf{D}_I) \approx \left(\frac{1}{2} + \epsilon\right)\frac{1}{2} + \frac{1}{2}\frac{1}{2} - \frac{1}{2} = \frac{1}{2}\epsilon$$

and the result follows. □

We emphasise that Theorems 1 and 4 mean that it is enough to prove that an undeniable or confirmer signature scheme has invisibility in the sense of Definition 2 to deduce both anonymity and invisibility in the sense of Camenisch and Michels [7].

5 Undeniable Signatures Based on RSA

Gennaro, Krawczyk and Rabin [15] described an undeniable/confirmer signature scheme based on RSA. In their case the signature for a message m is s where $s \equiv \overline{m}^d \pmod{N}$ and $\overline{m}$ is a one-way encoding. The signature may be verified by proving that $s^e \equiv \overline{m} \pmod{N}$ where the verification exponent e is known to the confirmer. The verification exponent e is fixed by publishing values $g = h^e \pmod{N}$. The original scheme required that the moduli be products of safe primes. The scheme was generalised to arbitrary RSA moduli by Galbraith, Mao and Paterson [14], who also gave a more efficient denial protocol.

To handle adaptive attacks on invisibility/anonymity it is clear that the one-way encoding must also be randomised. Hence, a signature becomes a pair (r, s) where r is random and $s \equiv H(m, r)^d \pmod{N}$ where $H(m, r)$ is the randomised one-way encoding.

5.1 Attacks on Invisibility and Anonymity

We show that the schemes of [14, 15] do not have invisibility or anonymity even under passive attacks. Note that these security properties were not claimed for either scheme.

Since d is odd it follows that the Jacobi symbols $(\frac{s}{N})$ and $(\frac{H(m,r)}{N})$ are equal. Hence, given a pair $(H(m,r), s)$ and a user's public key N, if $(\frac{s}{N}) \neq (\frac{H(m,r)}{N})$ then the signature is not valid for that user. This shows that the scheme does not have invisibility. A similar attack in the multi-user setting (testing the condition for all users' keys N_i) shows that the scheme does not have anonymity.

Another attack on anonymity of RSA-based schemes arises since all users must have different moduli N. If a signature s satisfies $s \geq N_i$ for some modulus N_i in the system then user i is not the signer of the message. This reduces the number of possibilities for the signer.

5.2 Preventing the Jacobi symbols Attack

To prevent the Jacobi symbols attack it is tempting to restrict to choices for r such that $H(m,r)$ is a quadratic residue in $\mathbb{Z}_N^*$ (and so s would also be a quadratic residue in $\mathbb{Z}_N^*$). This does not work since anonymity could still be broken by computing $(\frac{s}{N})$ and eliminating those for which the value is -1.

Another unsuccessful solution is the following. Since we do not want $(\frac{H(m,r)}{N})$ and $(\frac{s}{N})$ to be correlated a natural approach is to set $s = \xi H(m,r)^d \pmod N$ where ξ is a random element of order 2. The problem with this approach is that the confirmer (who has e) can recover ξ and hence factorise N and forge signatures.

Instead, our solution involves taking square roots, as these can be chosen to have arbitrary Jacobi symbol. We require that N is a Blum integer (i.e., a product $N = pq$ where $p \equiv q \equiv 3 \pmod 4$) and so for every $a \in \mathbb{Z}_N^*$ with $(\frac{a}{N}) = +1$ it follows that either a or $-a$ is a square. The idea is to obtain a value $H'(m,r)$ which depends on both $H(m,r)$ and N, and is such that $\pm H'(m,r)$ is a square modulo N. We define signatures by

$$s = \left(\sqrt{\pm H'(m,r)}\right)^d \pmod N$$

where the square-root is randomly chosen from among the four possibilities. The verification operation is to check that $s^{2e} \equiv \pm H'(m,r) \pmod N$ where the verification exponent e is only known to the signer and confirmer.

At first sight one might think this protocol to be insecure since two signatures s and s' on the same message leak a square-root of unity s/s'. However, this scenario never arises, as the value $H(m,r)$ is randomised.

We give further details on how $H'(m,r)$ is constructed. First we construct a one-way randomised padding $H(m,r)$ of the message using the method of Bellare and Rogaway [3]. One consequence of using a randomised padding scheme in the context of undeniable signatures is that it is necessary to transmit the value r as part of the signature. This is because, unlike with standard RSA signatures, the value $H(m,r)$ is not recovered by the verifier as part of the signature verification process.

We now must associate a value $H'(m,r)$ which is a square (or minus a square). To achieve this we insist that N be a Blum integer. We now provide a deterministic algorithm which takes $H(m,r)$ and N and produces an element $H'(m,r) \pmod N$ with Jacobi symbol $+1$. Let F be a hash function from k-bit strings to k-bit strings.[3] Set $i = 0$ and $v_0 = H(m,r)$. If $(\frac{v_i}{N}) \neq +1$ then iterate $v_{i+1} := F(v_i)$ and continue until an element v_t with Jacobi symbol $+1$ is obtained. Define $H'(m,r) = v_t$. This process is expected to terminate within a few iterations in practice.

[3] In the security proof we model F as a random oracle to ensure that the simulation is indistinguishable from a real game. In practice we believe that F could be the function $x \mapsto x + 1$ without loss of security.

5.3 Ensuring that Signature Length does not Reveal the Signer

A solution to this problem is for users to enlarge any values $s \pmod N$ to a fixed bitlength by adding a suitable multiple of N (see Section 3.1 of Desmedt [13] and also [4]). This padding removes any information about the size of N and does not interfere with the reduction of the value modulo N.

More precisely, let k be the bitlength of the modulus N. We will extend signatures to be bitstrings of length $2k$. Since N is an RSA modulus, s is indistinguishable from being uniformly distributed in $\mathbb{Z}_N$. Uniformly choose an integer in the range $0 \le t < (2^{2k} - s)/N$ and let $s' = s + tN$ (there are n or $n-1$ such t where $n = \lfloor (2^{2k}-1)/N \rfloor$). Then $0 \le s' < 2^{2k}$ and it follows that s' is indistinguishable from a random $2k$-bit string.

6 The New RSA-Based Scheme

We present the scheme in the case of moduli which are products of safe primes (a safe prime is a prime p such that $p' = (p-1)/2$ is also prime). The modification of the scheme to general Blum integer moduli is straightforward by following [14], but there are subtle reasons why our proof of invisibility does not apply in the general case (see the footnote in the proof of Theorem 5). The scheme is as follows for security parameter k.

System parameters: Let k be the security parameter and let k_0 be derived from k (e.g., $k_0 = \lfloor k/3 \rfloor$). The space $\mathcal{S}$ is the set of bitstrings of length $k_0 + 2k$. Hash functions G_0, G_1 and G_2 as in [3] must be specified. A hash function F which takes k-bit strings to k-bit strings must be specified.

Key generation: A signer chooses two primes p and q whose product is a k-bit integer, such that $p' = (p-1)/2$ and $q' = (q-1)/2$ are prime. The signer sets $N = pq$ and chooses $e, d \in \mathbb{Z}$ such that $ed \equiv 1 \pmod{\varphi(N)}$. The signer chooses $g \in \mathbb{Z}_N^*$ (such that, with overwhelming probability, $\{a^2 : a \in \mathbb{Z}_N^*\} \subseteq \langle g \rangle \subseteq \mathbb{Z}_N^*$) and sets $h = g^d \pmod N$.
The signer registers with the certificate authority with public key (N, g, h). The signer sends e to the designated confirmer via a secure channel (if there is one).

Signing: To sign a message m the signer constructs the randomised padding value $H(m,r)$ as in [3] (i.e., chooses a k_0-bit string r at random and computes $w = G_0(m,r), r^* = r \oplus G_1(w), \gamma = G_2(w)$ and $H(m,r) = w \| r^* \| \gamma$). Set $v = H(m,r)$. While $(\frac{v}{N}) \neq +1$ then set $v = F(v)$ and repeat. Set $H'(m,r) = v$. The signer computes

$$s = \left(\sqrt{\pm H'(m,r)}\right)^d \pmod N$$

where the sign is chosen so that $\pm H'(m,r)$ is a square modulo N and where the square-root is chosen randomly among the four possibilities. The signer enlarges s to a bitstring s' of length $2k$ by adding a suitable random multiple of N as in section 5.3. The signature on m is the $k_0 + 2k$ bit binary string (r, s').

Confirm/Deny: To confirm or deny a signature the signer or confirmer executes non-interactive, designated verifier versions of proofs[4] like those in [14] which prove knowledge of an integer e such that $g \equiv h^e \pmod N$ and $s^{2e} \stackrel{?}{\equiv} \pm H'(m,r) \pmod N$. Note that all users can compute $H'(m,r)$ given m, r and N.

Equivalence classes: Given a valid signature (r,s) then $(r, \pm s + tN)$ for values $t \in \mathbb{Z}$ such that $0 \le \pm s + tN < 2^{2k}$ is also a valid signature. This equivalence class is used in Definition 3. In case of confusion we stress that this equivalence class is smaller than $\{(r, \xi s + tN) : \xi^2 \equiv 1 \pmod N\}$ which is the set of all valid signatures for message m with randomness r.

[4] These proof transcripts must be encrypted when sent to the verifier if anonymity is to be preserved.

To obtain the security result it is necessary that executions of the confirm and deny protocols can be simulated in the random oracle model. This is not possible with interactive proofs so we must use non-interactive proofs. To maintain the security of the system (i.e., so that proofs cannot be transferred to other users) it is necessary to use designated-verifier proofs [16]. It is standard that such proofs can be simulated in the random oracle model. For further details see Jakobsson et al [16, 17].

7 Invisibility of Revised RSA-Based Undeniable and Confirmer Signatures

We now prove that the scheme described in the previous section has the invisibility property. Our proof only applies to the case of RSA moduli which are a product of safe primes. We write $\operatorname{ord}(g)$ for the order of an element modulo N (i.e., the smallest positive integer n such that $g^n \equiv 1 \pmod{N}$) and we write $\langle g_1, \ldots, g_m \rangle$ for the subgroup generated by $g_1, \ldots, g_m$.

We will show that the invisibility (and hence, anonymity) of the system depends on the hardness of the following computational problem:

Composite Decision Diffie-Hellman Problem (CDDH): Let N be a product of two safe primes (i.e., $N = pq$ with p and q both primes such that $p' = (p-1)/2$ and $q' = (q-1)/2$ are both prime). Consider the two sets

$$\mathcal{T} = \{(g, h, u, v) \in (\mathbb{Z}_N^*)^4 : \operatorname{ord}(g) = \operatorname{ord}(h) = 2p'q', h \in \langle g \rangle, \langle g, v \rangle = \mathbb{Z}_N^*\}$$

and

$$\begin{aligned}\mathcal{T}_{\text{CDDH}} = \{(g, h, u, v) \in \mathcal{T} : h &\equiv g^d \pmod{N} \text{ for some } d \text{ coprime to } \varphi(N),\\ v &\equiv \xi u^d \pmod{N} \text{ for some } \xi \in \mathbb{Z}_N^* \text{ of order } 2\}\end{aligned}$$

with the uniform distribution on each. The CDDH problem is to distinguish these two distributions. More precisely, a CDDH oracle with advantage ϵ is an algorithm $\mathcal{A}$ with input (N, g, h, u, v) such that

$$\begin{aligned}&\Pr\big(\mathcal{A}(N, g, h, u, v) = 1 \big| (g, h, u, v) \in \mathcal{T}_{\text{CDDH}}\big)\\ &- \Pr\big(\mathcal{A}(N, g, h, u, v) = 1 \big| (g, h, u, v) \in \mathcal{T}\big) = \epsilon.\end{aligned}$$

The CDDH assumption is that there is no CDDH oracle which runs in polynomial time and which has non-negligible advantage.

Belief in the CDDH assumption is given by the fact that if the factorisation of N is known then the problem reduces to the DDH problem modulo the prime factors of N.

Theorem 5. *Consider the RSA-based undeniable/confirmer signature scheme above with moduli which are products of safe primes. In the random oracle model (i.e., G_0, G_1, G_2 and F are random oracles) then the scheme has invisibility if the composite decision Diffie-Hellman problem is hard.*

Proof. Suppose we have an adversary $\mathbf{D}$ to the scheme. We will transform $\mathbf{D}$ into a CDDH algorithm $\mathcal{A}$. Let (N, g, h, u, v) be the input CDDH challenge problem to $\mathcal{A}$. Note that we can use random-self-reducibility of CDDH tuples if required (let a, b and c be coprime to the order of $\mathbb{Z}_N^*$ and let $g' = g^a, h' = h^a, u' = u^b g^c$ and $v' = v^b h^c$).

We execute $\mathbf{D}$ on the key (N, g, h). The distinguisher expects to perform hash queries, to obtain signatures on messages of its choice, and to run confirm and denial protocols. We now show how these will be simulated.

Hash query: A hash query could be with respect to any of the random oracles G_0, G_1, G_2 or F. We first analyse how to respond to a query $G_0(m, r)$ where m is a message and r is a random k_0-bit string.

If the value $G_0(m, r)$ has not been queried before then perform the following simulation algorithm:

- Choose x, y at random between 1 and N^2 and compute a k-bit string α which reduces modulo N to $\pm(g^x u^y)^2$.
  Store the values (x, y) as state information.
- Flip a coin.
- While tails do
  - Choose a random bitstring β of length k such that β corresponds to an element of $\mathbb{Z}_N^*$ with Jacobi symbol -1.
  - Define $F(\beta) = \alpha$ and store this as state information.
  - Set $\alpha = \beta$ and flip coin again.
- Parse the string α as $w \| r^* \| \gamma$, define $G_0(m, r)$ to be w, $G_1(w)$ to be $r \oplus r^*$, $G_2(w)$ to be γ and store all values as state information.

If there are any conflicts with existing definitions of the functions then the simulation algorithm halts (or retry with different (x, y)); this happens with negligible probability if k, k_0 and k_1 are sufficiently large.

The expected number of iterations of the while loop in the simulation algorithm is one.

A query on G_1, G_2 or F can be answered using the state information when the inputs are the result of a previous query on $G_0(m, r)$, and can be answered with a random bitstring otherwise.

Sign query: To sign m we choose a random r and construct $H'(m, r)$ using the simulation algorithm. The value for $H'(m, r)$ is an element of the form $\pm(g^x u^y)^2$ where (x, y) is known. Compute a bitstring of length $2k$ which reduces modulo N to

$$s = h^x v^y \pmod{N}.$$

The signature is (r, s).

For the challenge signature we choose the hidden bit b. If $b = 0$ we employ the above signing process. If $b = 1$ we choose s to be a random $2k$-bit string.

Confirm/Deny: First test whether the signature (r, s) is in the equivalence class of the challenge signature. If so then return an error message and halt, otherwise proceed.

It is necessary to decide whether the signature should be considered valid or not within the simulation. To do this, consider the hash value $H'(m, r) = \pm(g^x u^y)^2$ and check if $(s/(h^x v^y))^2 \equiv 1 \pmod{N}$ (if not then the signature is declared to be invalid). Note that the only signatures which are considered valid are signatures in the same equivalence class as signatures formed from a sign query (the probability that $\mathbf{D}$ constructs a valid signature any other way is negligible).

Once we have determined whether to respond positively or negatively then an appropriate proof can be simulated in the random oracle model.

The distinguisher $\mathbf{D}$ will eventually output a guess b' for the hidden bit b. If $b' = b$ then $\mathcal{A}$ outputs 1 (valid) as the answer to the CDDH problem and when $b' \neq b$ then $\mathcal{A}$ outputs 0 (invalid).

We now analyse the simulation. First, we claim that the simulated functions G_0, G_1, G_2 and F are indistinguishable from random oracles: The set of squares in $\mathbb{Z}_N^*$ is generated by g^2, hence every value $\alpha = \pm a^2$ for $a \in \mathbb{Z}_N^*$ can be written in the form $\pm(g^x u^y)^2$. Furthermore, the distribution of numbers of this form is indistinguishable from uniform if $1 \leq x, y \leq N^2$ (actually, we could take $1 \leq x, y \leq M$ for some smaller bound $M > N$ and still obtain a good result).

When the input tuple (g, h, u, v) lies in $\mathcal{T}_{\mathrm{CDDH}}$ then the Sign and Confirm/Deny oracles all behave perfectly. Hence, in this case, the simulation is identical to a real attack on the system, and so our CDDH algorithm $\mathcal{A}$ satisfies

$$\Pr(\mathcal{A}(N, g, h, u, v) = 1 | (g, h, u, v) \in \mathcal{T}_{\mathrm{CDDH}}) = \Pr(b' = b) = \frac{1}{2} + \mathrm{Adv}(\mathbf{D}).$$

We now consider the case where the input tuple is a random element of $\mathcal{T}$. In this case signatures generated by oracle queries are (with high probability) invalid. The simulation is therefore not indistinguishable from a real attack, and we cannot predict the outcome of the distinguisher $\mathbf{D}$. However, as we now show, the hidden bit b is independent of the game in this case.

Let $\alpha \in \mathbb{Z}_N^*$ be chosen arbitrarily such that $\pm\alpha = \beta^2$ is a square and let $s \in \mathbb{Z}_N^*$. We will show that one can choose integers $1 \leq x, y < N$ such that $\alpha = \pm(g^x u^y)^2$ and $s = h^x v^y$. First we consider the projection of all values into the cyclic subgroup of squares in $\mathbb{Z}_N^*$. By abuse of notation we may write $h = g^d, u = g^{c_1}, v = g^{c_2}, \beta = g^{c_3}$ and $s = g^{c_4}$. Solving the equations $\alpha = \pm(g^x u^y)^2$ and $s = h^x v^y$ is equivalent to solving

$$\begin{pmatrix} 1 & c_1 \\ d & c_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \equiv \begin{pmatrix} c_3 \\ c_4 \end{pmatrix} \pmod{p'q'}.$$

The matrix is invertible when $c_2 - dc_1$ is coprime to $p'q'$ and this happens with overwhelming probability[5] when (g, h, u, v) is chosen uniformly at random from $\mathcal{T}$. Finally, we consider the squares: The value of α places no restriction on elements of order 2 in β, and h and v together generate $\mathbb{Z}_N^*$ so we can solve the projection of the equation $s = h^x v^y$ in the subgroup of elements of order 2.

This proves that the hidden bit is independent of the simulation when the input (g, h, u, v) is a random 4-tuple. Hence the output of $\mathcal{A}$ in this case differs negligibly from a coin-toss.

It follows that

$$\mathrm{Adv}(\mathcal{A}) \approx \frac{1}{2} + \mathrm{Adv}(\mathbf{D}) - \frac{1}{2} = \mathrm{Adv}(\mathbf{D})$$

(where $\approx$ means equality up to negligible factors). □

Corollary 1. *The RSA-based undeniable/confirmer signature scheme above has anonymity in the random oracle model.*

Note that it remains an open problem to obtain invisibility and anonymity in the case of moduli which are not necessarily safe prime products.

8 Other Security Properties

We now prove the security against forgery of the RSA-based undeniable signature scheme presented in Section 5.

8.1 Unforgeability

We show that our scheme resists forgery in the random oracle model if factoring is hard (this is much easier than the equivalent result for the Chaum and van Antwerpen scheme, which is due to Okamoto and Pointcheval [20]). Our proof handles both the case where the adversary is a designated confirmer (i.e., the adversary knows the decryption exponent e) and the general case.

Theorem 6. *In the random oracle model, if factoring integers which are products of safe primes is hard then the RSA-based scheme is secure.*

[5] This is where the difficulties arise if N is not a product of safe primes.

Proof. Let $\mathcal{F}$ be an adversary which forges an RSA-based undeniable signature for our scheme in a CPA, CCA1 or CCA2 attack scenario. We show how to build a factoring algorithm using $\mathcal{F}$.

Let N be the challenge integer and choose a random odd integer e (since N is a product of safe primes then e is coprime to $\varphi(N)$ with overwhelming probability).[6] Choose a random $t \in \mathbb{Z}_N^*$ and define $g = t^e \pmod N$ and $h = t$. The input to $\mathcal{F}$ is the triple (N, g, h) and if the forger is the confirmer then they are also given e.

The forger $\mathcal{F}$ will make various queries to the simulation and this is how we respond to them:

Hash query These are answered similarly to the proof of Theorem 5. The basic step is to define $H'(m, r)$ as follows: Choose a random element $a \in \mathbb{Z}_N^*$ and set $H'(m, r) \equiv \pm a^{2e} \pmod N$.

Sign query: If $\mathcal{F}$ asks for an undeniable signature on m then construct $H'(m, r) = \pm a^{2e}$. The response is the pair (r, a). Since a was chosen randomly then it has random Jacobi symbol.

Confirm/Deny: We determine the validity of a given signature using the verification exponent e and then simulate the proof accordingly in the random oracle model.

One can easily show that this simulation is indistinguishable from a real game. Eventually the forger outputs a triple (m, r, s) such that

$$s^{2e} \equiv \pm H'(m, r) \pmod N.$$

Since N is a Blum integer and s^{2e} is a square then we have a minus in the above formula if and only if $H'(m, r)$ was defined to be $-a^{2e}$. Hence we have $s^{2e} \equiv a^{2e} \pmod N$ and so

$$(s^e/a^e)^2 \equiv 1 \pmod N.$$

Since a was chosen randomly then, with probability $1/2$, we have a nontrivial root of unity which gives a factorisation of N. □

8.2 Convertibility

In [14, 15] it is shown how to obtain an undeniable/confirmer signature scheme based on RSA which allows conversion to ordinary RSA signatures. The same approach applies to our scheme.

The basic idea is to have values e, d and c such that $edc \equiv 1 \pmod{\varphi(N)}$. The value e is now public, while d is known only to the signer and c is used for confirming/converting. The scheme of Section 5 is now verified using the relation $s^{2ce} \stackrel{?}{\equiv} \pm H'(m, r) \pmod N$. Signatures may be selectively converted to public key signatures by raising to the power c (this should be accompanied by a zero-knowledge proof of correctness). All signatures may be converted by publishing the value c.

9 Previous Schemes

For completeness, we discuss the invisibility and anonymity properties of some of the existing schemes in the literature. Of course, the first two schemes were not developed to have invisibility and are already known to not have this property.

1. The undeniable signature scheme of Chaum and van Antwerpen [9, 10].
2. The Chaum confirmer signature scheme [12].
3. The schemes of Michels and Stadler [18].
4. The scheme of Camenisch and Michels [7].

[6] To generalise this result to moduli which are not products of safe primes it suffices to choose e to be a reasonably large prime.

9.1 Chaum and van Antwerpen Scheme

The original undeniable signature scheme of Chaum and van Antwerpen [9] is as follows. Let $g \in \mathbb{F}_p^*$ have prime order. Each user has a secret key x and a public key $h = g^x$. The undeniable signature on message m is $s = m^x$. To confirm the signature one must interact with the signer and perform an interactive proof of equality of discrete logarithms on the tuple (g, h, m, s).

The problem of determining if a signature was generated using a certain user's public key is essentially the decision Diffie-Hellman (DDH) problem. We recall the DH and DDH problems (in a slightly more general form than usual):

Definition 5. *Let G be a group The Diffie-Hellman problem (DH) is the problem, over random values for $g \in G$, $1 \leq a \leq \#G$ and $h \in G$, of computing h^a given (g, g^a, h). The Decision-Diffie-Hellman problem (DDH) is the problem of distinguishing the two distributions on G^4 given by (g, g^a, h, h^a) and (g, g^a, h, h^b) for $1 \leq a, b \leq \#G$.*

As is well known, the DDH problem is not necessarily hard in groups whose order has small prime factors. If users allow m to be such that $(\frac{m}{p}) = -1$ then, by considering the Legendre symbol of a signature s which is known to be valid, one can determine whether the secret key x of a given user is even or odd. Once the parity of x is known one can distinguish between a valid signature (g, g^x, m, m^x) and a random tuple (g, g^x, m, s) about 25% of the time (which is non-negligible), by comparing the Legendre symbols $(\frac{m}{p})$ and $(\frac{s}{p})$. Hence the scheme does not have invisibility. Similarly, the scheme does not have anonymity.

In Sections 10 and 11 we show how to obtain a secure undeniable signature scheme by simple modifications to the original Chaum and van Antwerpen scheme.

9.2 Chaum Confirmer Scheme

The Chaum confirmer scheme [12] uses a discrete logarithm system with generator g and uses RSA signatures. The public key of the signer is (N, e). The confirmer has public key $h = g^x$. The signer chooses a random r, computes $a = g^r, b = h^r$ and $\alpha = (H(a, b) \oplus F(m))^d \pmod{N}$ (where F is a hash function and H is an invertible mixing function). The signature is (a, b, α). Both the signer and the confirmer are able to prove to another user that the signature is valid (see [12] for details).

In this scheme the signature includes $\alpha \in \mathbb{Z}_N^*$ such that $\alpha^e \equiv m \pmod{N}$ where m is known (it is a function of the signature components a and b) and where (N, e) is the public key for a user. If the Jacobi symbols $(\frac{\alpha}{N})$ and $(\frac{H(a,b) \oplus F(m)}{N})$ are not equal then the signature is invalid, and so this scheme does not have anonymity.

9.3 Michels and Stadler Scheme

Michels and Stadler [18] give two solutions, both using the tool of 'confirmer commitments'. The first uses 3-move zero knowledge proofs for signature while the second uses existentially forgeable signature schemes (such as RSA signatures).

The schemes of Michels and Stadler claim to have the invisibility property (with respect to a weaker definition than the one we have presented).

The RSA-based scheme of Section 5.4 of Michels and Stadler [18] does not have the anonymity property. In this scheme the signature is a usual RSA signature of a known value m (which is $h_B(d) + b$ in the setting of [18], where all quantities are visible to an adversary). Hence an attack using the Jacobi symbol can be applied.

We believe that the protocol in [18] based on Schnorr signatures can be shown to have the invisibility and anonymity properties if minor modifications (such as in Section 10) are made.

9.4 Camenisch and Michels Scheme

Camenisch and Michels [7] give a general construction to build a confirmer signature scheme from a signature scheme and an encryption scheme. Let the confirmer have a public key K_C for the encryption scheme and a signer have a public key K_S for the signature scheme. To sign a message m the signer computes $s = \mathrm{Sign}(m)$ and $e = \mathrm{Enc}(s, K_C)$ and publishes e as their signature. The confirmer can decrypt e to obtain s and thus determine the validity of the signature using K_S. The confirmer is able to prove the validity to other entities using a zero-knowledge proof, which in general requires binary challenges and is very inefficient. Our solutions are much more efficient than the methods of Camenisch and Michels.

If one uses techniques such as those developed in this paper then one can build a version of the confirmer signature scheme of [7] which has Property A. It follows from the results of [7] and Theorems 2 and 4 that this scheme has anonymity.

10 Undeniable Signatures in Finite Fields

We have seen how RSA-based schemes can provide anonymity even though each user is working in a different group. This opens the possibility that the scheme of Chaum and van Antwerpen could also be developed in a situation where users do not share the same finite field. In this section we show how to achieve this without any loss of security.

The first step is obviously to ensure that the attack described in section 9.1 cannot be applied. This attack relied on the presence of elements of order two, but more general versions can be applied using elements of any small order, since the factorisation of $p-1$ is assumed to be public. Clearly, the attack also generalises from prime fields $\mathbb{F}_p$ to more general finite fields $\mathbb{F}_q$ where $q = p^n$. We will present the scheme in terms of a general finite field $\mathbb{F}_q$.

There are two ways to proceed, one is to 'blind' the signature using elements of small order. Since the factorisation of $q-1$ is known (and is shared by all users) it is equivalent to work in a subgroup of large prime order l of the finite field $\mathbb{F}_q^*$ (in this case all elements will have Legendre symbol $+1$). Since these two formulations are equivalent one might think that either can be adopted. However, consider the following attack on the latter formulation: Suppose a user's signature is a bitstring corresponding to an element of order l in a finite field $\mathbb{F}_q$. If two users have different fields $\mathbb{F}_{q_1}^*$ and $\mathbb{F}_{q_2}^*$ with corresponding primes l_1 and l_2 then it is easy to determine whether a bitstring s corresponds to an element of order l_i in $\mathbb{F}_{q_i}^*$ or not, and so the anonymity of the scheme can be broken.

Hence, our basic approach is as follows. Let each user choose a prime (or prime power) q which is at most k bits long. Write $q-1 = nl$ where l is a (large) prime and where n is some cofactor. Let $g \in \mathbb{F}_q^*$ have order l and let $h = g^x$. The public key for a user is (q, n, l, g, h) and the secret key is x. The signature on $H(m, r)$ is a bitstring of length $2k$ such that s 'reduces' to an element

$$\xi H(m, r)^x \in \mathbb{F}_q$$

where $\xi \in \mathbb{F}_q^*$ is a random element of order dividing n. The word 'reduces' may be taken to be usual reduction modulo q when q is a prime. In other words, the string s is obtained by adding a random multiple of q to the residue $\xi H(m, r)^x \pmod{q}$ just as was done in the RSA case. When q is a prime power then natural generalisations of this approach may be used, depending on the representation used for finite field elements. The equivalence class of a signature s in this case (see Definition 3) is the set of bitstrings of length $2k$ which reduce (depending on the rule for turning bitstrings into finite field elements) to $\xi s \in \mathbb{F}_q$ where ξ has order dividing n.

We summarise the revised scheme for security parameter k:

System parameters: Let $\mathcal{M}$ be a space of binary strings and let $\mathcal{S}$ be the space of $(k_0 + 2k)$-bit binary strings. Let $\mathcal{K}$ be the space of all (q, n, l, g, h) where q is a k-bit prime power, $q - 1 = nl$ where l is a prime of at least $ck^{1/3}$ bits for some constant c (e.g., $c = 16$) and where g and h have order l in $\mathbb{F}_q^*$. Let G_0, G_1, G_2 be hash functions as in [3].

Gen: Choose a k-bit prime power q such that $(q-1)$ is divisible by some large prime l. We insist that $l^2 \not| (q-1)$. Let $n = (q-1)/l$. Choose an element $g \in \mathbb{F}_q^*$ whose order is l, choose a random secret x, and compute $h = g^x$. The public key is (q, n, l, g, h).

Sign: The signature on a message m for public key $h = g^x$ is a pair (r, s') where s' is a $2k$-bit binary string which reduces to a finite field element $s = \xi H(m,r)^x \in \mathbb{F}_q$ where $\xi \in \mathbb{F}_q^*$ is an element of order dividing n and where $H(m,r)$ is the padding scheme from [3] as used above.

Confirm/Deny: These are proofs of equality/inequality of discrete logarithms in the subgroup of $\mathbb{F}_q^*$ of order l. For details see below.

Equivalence classes: The equivalence classes (see Definition 3) in this case are all pairs (r, s'') where s'' reduces to ξs and where ξ runs over elements of order dividing n.

It is likely that the scheme could also be developed for other groups for which the DDH problem is thought to be hard (e.g., elliptic curves over finite fields). The difficulties would be hashing to obtain group elements and lifting group elements to give binary strings which are indistinguishable from random.

For the sake of completeness we give the signature confirmation protocol of [16] for the challenge $(g, h, H(m,r), s)$ where $g, h, H(m,r)$ and s lie in $\mathbb{F}_q^*$ and where $h = g^x$ is the public key of the signer. The first step is to raise all elements to the power $(q-1)/l$ so that they all lie in the subgroup of order l (we use the same notation for these elements). We also use the public key $y \in \langle g \rangle$ of the designated verifier (this is for the trapdoor commitment scheme).

1. Prover chooses random $w, r, t \in \mathbb{Z}_l$ and computes $c = g^w y^r, G = g^t, M = H(m,r)^t$, $b = H'(c, G, M)$ and $d = t + x(b+w) \pmod{l}$ where H' is some cryptographically strong hash function with full domain output onto $\mathbb{Z}_l$.
2. Prover sends (w, r, G, M, d) to the verifier[7].
3. Verifier computes $c = g^w y^r$ and $b = H'(c, G, M)$ and checks that $Gh^{b+w} = g^d, Ms^{b+w} = H(m,r)^d$.

This protocol is sound, complete and zero knowledge in the random oracle model and if the discrete logarithm problem is hard. We refer to [16] for the further discussion of the security of this protocol. We note that it is implicit in the confirmation protocol that the verifier knows the value $H(m,r)$. Hence the value r must be transmitted to the verifier.

For signature denial we use the protocol of Jakobsson [17] which essentially requires two executions of the confirmation protocol.

We note that the revised scheme has all the other desirable security properties of an undeniable signature scheme (see Chaum et al [9, 10], Camenisch and Michels [7] and Okamoto and Pointcheval [20]). In particular, the methods of [20] allow a proof that forgery is hard in the random oracle model if the Diffie-Hellman problem in the subgroup of order l is hard relative to a Decision-Diffie-Hellman oracle.

11 Invisibility of Finite Field Based Scheme

The revised Chaum undeniable signature scheme clearly avoids the attack on anonymity mentioned earlier since elements of small order give no information about the signature. We now prove the invisibility of this scheme under the assumption that the decision Diffie-Hellman problem (DDH) in the subgroup of $\mathbb{F}_q^*$ of large prime order l is hard.

Theorem 7. *Let $\mathcal{U}$ be the revised Chaum undeniable signature scheme. Suppose that, for values of the security parameter k, the Decision Diffie-Hellman problem in the subgroup of elements of order l in $\mathbb{F}_q^*$ is hard. Then, in the random oracle model, $\mathcal{U}$ has the invisibility property under an adaptive chosen message attack.*

[7] This is a non-interactive, designated-verifier proof. However, verification requires the public key h of the prover. Hence the prover must also send a public key certificate. If online eavesdropping is a serious threat then the certificate must be encrypted, as it contains the identity of the signer.

Proof. Suppose that an adversary **D** to the undeniable signature scheme exists. We will transform it into a DDH algorithm. Let the input DDH problem be (g_1, g_2, g_3, g_4) (we can apply random-self-reducibility of Diffie-Hellman tuples if required).

We first set up the public key (g_1, h_1) and run the adversary **D** on this public key pair. The adversary will expect to consult hash and sign oracles and will also expect to engage in runs of the confirmation and denial protocols. At some point **D** will produce a message m and request a challenge undeniable signature from one of the two public keys. We must simulate all these operations.

Hash query: When the adversary **D** makes a hash query on m with randomness r we check whether $H(m, r)$ has already been defined. If not, then random integers x and y are chosen and, as in the proof of Theorem 5, $H(m, r)$ is defined to be $\xi g_1^x g_3^y$ for some random $\xi \in \mathbb{F}_q^*$ of order dividing $(q-1)/l$.

Sign query: When **D** makes a sign query with a message m we first ensure that $H(m, r)$ is defined and obtain the matching values of x and y. Output $s = \xi g_2^x g_4^y$ for some random ξ of order dividing $(q-1)/l$.

In the case of the challenge message we choose the bit b at random. If $b = 0$ then we respond with a signature as above and if $b = 1$ then we respond with a random element of $\mathbb{F}_q^*$.

Confirm/Deny: When the adversary wants to engage in a signature confirmation or denial on (m, r, s) with respect to public key i we first have to determine whether the signature is valid or not.

Within the simulation we can do this by determining the value $H(m, r) = \xi g_1^x g_3^y$ (choosing x and y at random if the query has not been made earlier). Declare the signature s to be valid if and only if $s/(g_2^x g_4^y)$ has order dividing $(q-1)/l$. Note that the probability that **D** can construct a valid signature without querying the sign oracle is negligible.

Once the validity (within the simulation) of the signature has been determined we know whether to respond positively or negatively to the execution of the confirmation or denial protocol. Since the zero knowledge proofs are perfectly simulatable in the random oracle model we can easily construct a proof which gives a suitable response.

Finally, the adversary will output its guess b' to the value of the bit b. If $b = b'$ then output the result 'true' for the validity of the Diffie-Hellman tuple, and if $b \neq b'$ then output 'false'.

When the input is a valid Diffie-Hellman tuple then the simulation is identical to a genuine attack on the cryptosystem. Hence the advantage for the Decision-Diffie-Hellman algorithm is exactly the same as the advantage of **D**.

When the input is not a valid Diffie-Hellman tuple then the simulation is not valid. However, the transcript of values for $H(m, r)$ and s is indistinguishable from a uniform distribution (since for all $u, v \in \langle g_1 \rangle$ there is some x, y such that $g_1^x g_3^y = u$ and $g_2^x g_4^y = v$). Hence the transcript is independent of the hidden bit b. This means that the adversary **D** has no chance of correctly guessing the signer and the probability that $b = b'$ is $1/2$. This argument includes the case where **D** detects that the simulation is invalid. Regardless of what strategy is used by **D** in this case the probability that $b = b'$ is $1/2$.

It is clear that the DDH adversary has essentially the same performance as **D**. It is standard to transform this DDH algorithm into one which outputs 'valid' and 'invalid' as answers to the DDH problem with any desired accuracy. □

12 Anonymity Between Signatures of Different Schemes

We have shown that our RSA-based scheme has the invisibility property when $\mathcal{S}$ is the space of binary strings of length $k_0 + 2k$. We have also given a finite field based scheme which has invisibility with respect to the same signature space.

Theorem 4 implies that a given undeniable/confirmer signature scheme has anonymity in the case of two users. In fact, the proof of Theorem 4 relies only on the fact that both users share the same signature space $\mathcal{S}$. One can easily modify Definitions 3 and 4 and the statement of Theorem 4 so that they only reference $\mathcal{S}$ and not the precise scheme. Hence we obtain anonymity in the more general case where one user has an RSA scheme and the other a finite field scheme. In other words, we find ourselves in the strong position of preserving anonymity even when different users base their systems on quite different public key mechanisms.

13 Acknowledgements

The authors would like to thank Simon Blackburn, Markus Jakobsson, Arjen Lenstra and Kenny Paterson for helpful comments. The authors also thank an anonymous referee for pointing out a weakness in an earlier version of the paper.

References

1. G. Ateniese and G. Tsudik, Some open issues and directions in group signatures, in M Franklin (ed.) FC '99, Springer LNCS 1648 (1999) 196–211.
2. G. Ateniese, D. Song and G. Tsudik, Quasi-efficient Revocation in Group Signatures, to appear in M. Blaze (ed.), Financial Cryptography 2002, Springer LNCS.
3. M. Bellare and P. Rogaway, The exact security of digital signatures - how to sign with RSA and Rabin, in U. Maurer (ed.), EUROCRYPT '96, Springer LNCS 1070, (1996) 399–416.
4. M. Bellare, A. Boldyreva, A. Desai and D. Pointcheval, Key-privacy in public key encryption, in C. Boyd (ed.) ASIACRYPT 2001, Springer LNCS 2248 (2001) 566–582.
5. E. Bresson and J. Stern, Efficient revocation in group signatures, in K. Kim (ed.), PKC 2001, Springer LNCS 1992 (2001) 190–206.
6. J. Camenisch and M. Stadler, Efficient group signature schemes for large groups, in B. Kaliski (ed.) CRYPTO '97, Springer LNCS 1294 (1997) 410–424.
7. J. Camenisch and M. Michels, Confirmer signature schemes secure against adaptive adversaries, in B. Preneel (ed.), EUROCRYPT 2000, Springer LNCS 1870 (2000) 243–258.
8. J. Camenisch and A. Lysyanskaya, Dynamic accumulators and application to efficient revocation of anonymous credentials, in M. Yung (ed.), CRYPTO 2002, Springer LNCS 2442 (2002) 61–76.
9. D. Chaum and H. van Antwerpen, Undeniable signatures, in G. Brassard (ed.), CRYPTO '89, Springer LNCS 435 (1990) 212–216.
10. D. Chaum, Zero-knowledge undeniable signatures, in I.B. Damgaard (ed.), CRYPTO '90, Springer LNCS 473 (1991) 458–464.
11. D. Chaum, E. van Heijst and B. Pfitzmann, Cryptographically strong undeniable signatures, unconditionally secure for the signer, in J. Feigenbaum (ed.), CRYPTO '91, Springer LNCS 576 (1992) 470–484.
12. D. Chaum, Designated confirmer signatures, in A. de Santis (ed.), EUROCRYPT 94, Springer LNCS 950 (1995) 86–91.
13. Y. Desmedt, Securing traceability of ciphertexts: Towards a secure software escrow scheme, in Guillou et al. (eds.), EUROCRYPT '95, Springer LNCS 921 (1995) 147–157.
14. S. D. Galbraith, W. Mao, and K. G. Paterson, RSA-based undeniable signatures for general moduli, in B. Preneel (ed.), Topics in Cryptology – CT-RSA 2002, Springer LNCS 2271 (2002) 200–217.
15. R. Gennaro, H. Krawczyk and T. Rabin, RSA-based undeniable signatures, in W. Fumy (ed.), CRYPTO '97, Springer LNCS 1294 (1997) 132–149.
    Also in *Journal of Cryptology* (2000)13:397–416.
16. M. Jakobsson, K. Sako and R. Impagliazzo, Designated verifier proofs and their applications, in U. Maurer (ed.) EUROCRYPT '96, Springer LNCS 1070 (1996) 143–154.
17. M. Jakobsson, Efficient oblivious proofs of correct exponentiation, in B. Preneel (ed.), Communications and multimedia security, Kluwer (1999) 71–84.
18. M. Michels and M. Stadler, Generic constructions for secure and efficient confirmer signature schemes, in K. Nyberg (ed.) EUROCRYPT '98, Springer LNCS 1403 (1998) 406–421.
19. T. Okamoto, Designated confirmer signatures and public key encryption are equivalent, in Y. G. Desmedt (ed.), CRYPTO '94, Springer LNCS 839 (1994) 61–74.

20. T. Okamoto and D. Pointcheval, The Gap-problems: a new class of problems for the security of cryptographic schemes, in K. Kim (ed.) PKC '2001, Springer LNCS 1992 (2001) 104–118.
21. K. Sakurai and S. Miyazaki, A bulletin-board based digital auction scheme with bidding down strategy - towards anonymous electronic bidding without anonymous channels nor trusted centers, In Proc. International Workshop on Cryptographic Techniques and E-Commerce, City University of Hong Kong Press, (1999) 180–187.
22. K. Sakurai and S. Miyazaki, An anonymous electronic bidding protocol based on a new convertible group signature scheme, in E. Dawson et al. (eds.), ACISP 2000, Springer LNCS 1841 (2000) 385–399.