Centrum voor Wiskunde en Informatica
REPORT*RAPPORT*

A note on parallel executions of restrictive blind issuing protocols for secret-key certificates

S.A. Brands

Computer Science/Department of Algorithmics and Architecture

CS-R9519 1995

Report CS-R9519
ISSN 0169-118X

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

A Note on Parallel Executions of Restrictive Blind Issuing Protocols for Secret-Key Certificates

Stefan Brands

CWI

P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Abstract

The receiver in a restrictive blind certificate issuing protocol can blind the issued public key and the certificate but not a certain blinding-invariant predicate of the secret key. Recently a generally applicable technique was described for designing restrictive blind issuing protocols for a certain type of secret-key certificates, and it was shown that the resulting issuing protocols should not be run in parallel since that would enable an attack in which completely blinded triples can be retrieved. To allow the signer in highly demanding applications to run the restrictive blind certificate issuing protocol in parallel mode, a simple technique for modifying the issuing protocols was proposed. In this note it is shown that the proposed modification technique does not overcome the parallel attack problem.

AMS Subject Classification (1991): 94A60
CR Subject Classification (1991): D.4.6
Keywords and Phrases: Cryptography, Certificates.

1. Introduction

Recently [2] the notion of secret-key certificates was introduced. These certificates can be used for secure management of cryptographic keys in much the same way as can public-key certificates. Of particular interest for privacy-protecting mechanisms for signature transport are so-called restrictive blind certificate issuing protocols, introduced in [1]; the receiver in such a protocol can blind the issued public key and the certificate but not a certain blinding-invariant predicate of the secret key. In [3] a generally applicable technique was described for designing restrictive blind secret-key certificate issuing protocols, and it was shown that the technique can be applied to any Fiat-Shamir type signature scheme that can be turned into an ordinary blind signature scheme by applying a divertability technique due to Okamoto and Ohta [11]; no such generally applicable technique has yet been proposed for public-key certificates.

Building on this work, it was shown in [4] that any of the new restrictive blind certificate issuing protocols can be used to design an efficient off-line untraceable electronic cash system. In particular, at least any of the following Fiat-Shamir type signature schemes can be used as the basis of an off-line untraceable cash system: Guillou-Quisquater [9], Schnorr [12], Fiat-Shamir [8], Brickell-McCurley [6], Feige-Fiat-Shamir [7], and Okamoto [10] (several schemes). Given any of these Fiat-Shamir type signature schemes, one merely needs to apply mechanically the techniques described in [2] (for deriving a secret-key certificate scheme from the Fiat-Shamir signature scheme), [3] (for designing a restrictive blind issuing protocol for the resulting secret-key certificate scheme), and [4] (for building the cash system protocols around the restrictive blind certificate issuing scheme). The description in [4] demonstrates this process explicitly for the Schnorr scheme, and the construction based on the Guillou-Quisquater scheme then follows trivially from the description in [3].

As shown in [3] the restrictive blind issuing protocols may not be run in parallel, because that would enable attackers to retrieve certified key pairs for which even the presumed blinding-invariant part of the secret keys can be blinded. To enable the signer in highly demanding applications to securely run executions of the proposed restrictive blind certificate issuing protocols in parallel, a modification was proposed in [3] (see also [4]). Since the provided argument in favour of its presumed security under parallel executions considers only one particular type of attack, the reader was urged to be cautious and encouraged to try to break the modified issuing protocol in other ways.

In this note it is shown that the proposed modification indeed is ineffective to overcome the parallel attack problem.

2. Why the Proposed Modification is Ineffective.

Since the description in [3] is in terms of the Guillou-Quisquater scheme, the alternative attack in parallel mode will be described in terms of that scheme first.

In the restrictive blinding issuing protocol for the Guillou-Quisquater secret-key certificate scheme, as described in [3], a signer $\mathcal{S}$ uses a public key $(n, v, h, g, \mathcal{H}(\cdot))$ and a corresponding secret key $(h^{1/v}, g^{1/v})$, denoted by (x, y) (alternatively, and more powerful, the factorization of n can serve as the secret key). Here, n denotes the product of two distinct prime numbers, v is a prime number that is co-prime with $\varphi(n)$ (although other choices can be used as well, such as for instance taking v to be twice

such a prime), h and g are elements of $\mathbb{Z}_n^*$, and $\mathcal{H}(\cdot)$ is a polynomial-size description of a correlation-free one-way hash-function that maps its inputs to $\mathbb{Z}_{2^t}$ for some appropriate t. A *secret-key certificate* of $\mathcal{S}$ on a public key h_i of a receiver $\mathcal{R}_i$ is a pair $(r, c) \in \mathbb{Z}_n^* \times \mathbb{Z}_{2^t}$ such that

$$c = \mathcal{H}(h_i, r^v (h\, h_i)^{-c}).$$

A secret key of $\mathcal{R}_i$ corresponding to its public key h_i is a pair (s_{0i}, s_{1i}) such that

$$h_i = g^{s_{0i}} s_{1i}^v.$$

(For off-line electronic cash purposes $\mathcal{R}_i$ must use a one-time key pair (h_i, a_i) instead of h_i, as shown in [4].)

In an execution of the issuing protocol $\mathcal{R}_i$ receives a certified key pair (s_{0i}, s_{1i}), h_i', (r', c'), with the blinding-invariant predicate of the secret key being equal to $s_{0i} \bmod v$. The number $g^{s_{0i}}$ will be denoted by h_i; it can be thought of as the "not-yet-blinded" public key that is to be blinded to h_i'. The issuing protocol is as follows:

Step 1. $\mathcal{S}$ generates at random a number $w \in \mathbb{Z}_n^*$, and sends $a := w^v$ to $\mathcal{R}_i$.

Step 2. $\mathcal{R}_i$ generates at random two numbers $s_{1i}, t_1 \in \mathbb{Z}_n^*$, and a number $t_2 \in \mathbb{Z}_v$. $\mathcal{R}_i$ computes $h_i' := h_i s_{1i}^v$, $c' := \mathcal{H}(h_i', t_1^v (h\, h_i)^{t_2} a)$, and sends $c := c' + t_2 \bmod v$ to $\mathcal{S}$.

Step 3. $\mathcal{S}$ sends $r := (x y^{s_{0i}})^c w$ to $\mathcal{R}_i$.

$\mathcal{R}_i$ accepts if and only if $r^v (h\, h_i)^{-c} = a$. If this verification holds, $\mathcal{R}_i$ computes $r' := r\, t_1 (h\, h_i)^{c' + t_2 \operatorname{div} v} s_{1i}^{c'}$.

As shown in Section 3.6 of [3], this issuing protocol is not restrictive blind when executions with respect to different blinding-invariant numbers can be performed in parallel. Motivated by the fact that the particular attack described in [3] requires the attacking receivers to compute their respective challenges in terms of their respective blinding-invariant numbers, a modification technique was proposed in [3] that simply consists of ensuring that $\mathcal{R}_i$ in the modified issuing protocol learns s_{0i} only *after* having sent his challenge.

It will now be shown that this modification is ineffective, since an alternative attack, similar to that in Section 3.6 of [3], can be performed that does *not* require knowledge of the respective blinding-invariant numbers beforehand.

Using the notation of Section 3.6 of [3], let s_{0i} be the blinding-invariant number for $\mathcal{R}_i$, and $s_{0j} \neq s_{0i} \bmod v$ that for $\mathcal{R}_j$; the corresponding "not-yet-blinded" public keys are h_i and h_j. In its simplest form, the alternative attack on two parallel executions of the issuing protocol is the following:

(Step 1 for $\mathcal{R}_i$) $\mathcal{S}$ generates at random a number $w_i \in \mathbb{Z}_v$, and sends $a_i := w_i^v$ to $\mathcal{R}_i$.

(Step 1 for $\mathcal{R}_j$) $\mathcal{S}$ generates at random a number $w_j \in \mathbb{Z}_v$, and sends $a_j := w_j^v$ to $\mathcal{R}_j$.

(Cooperation between $\mathcal{R}_i$ and $\mathcal{R}_j$) $\mathcal{R}_i$ and $\mathcal{R}_j$ decide on an arbitrary number $\alpha \in \mathbb{N}$, and compute $h_k := h_i^{\alpha} h_j^{1-\alpha}$. They then compute $c_k := \mathcal{H}(h_k, a_i a_j)$.

(Step 2 for $\mathcal{R}_i$) $\mathcal{R}_i$ sends $c_i := \alpha c_k \bmod v$ to $\mathcal{S}$.

(Step 2 for $\mathcal{R}_j$) $\mathcal{R}_j$ sends $c_j := (1-\alpha) c_k \bmod v$ to $\mathcal{S}$.

(Step 3 for $\mathcal{R}_i$) $\mathcal{S}$ sends $r_i := (x y^{s_{0i}})^{c_i} w_i$ to $\mathcal{R}_i$.

(Step 3 for $\mathcal{R}_j$) $\mathcal{S}$ sends $r_j := (x y^{s_{0j}})^{c_j} w_j$ to $\mathcal{R}_j$.

$\mathcal{R}_i$ and $\mathcal{R}_j$ accept if and only if

$$r_i^v (h\, h_i)^{-c_i} = a_i \quad \text{and} \quad r_j^v (h\, h_j)^{-c_j} = a_j.$$

If this verification holds, then $\mathcal{R}_i$ and $\mathcal{R}_j$ compute

$$r_k := r_i r_j (h h_i)^{-(\alpha c_k \operatorname{div} v)} (h h_j)^{-((1-\alpha) c_k \operatorname{div} v)}.$$

Proposition 1 *Let s_{0k} denote $\alpha s_{0i} + (1-\alpha) s_{0j}$. If $\mathcal{R}_i$ and $\mathcal{R}_j$ accept, then*

$$(s_{0k}, 1),\ h_k,\ (r_k, c_k)$$

is a certified key pair.

Since the proof is similar to that of Proposition 9 in [3] (full paper), it is omitted here.

Observe that the attackers in the above attack do not need to know their respective blinding-invariant numbers s_{0i} and s_{0j} in order to determine c_i and c_j (they only need to learn them at some time or another in order to know the secret key $(s_{0k}, 1)$ corresponding to h_k), which immediately shows that the proposed modification in Section

3.6 of [3] is ineffective: it is based on the incorrect assumption that the attackers must know their respective blinding-invariant numbers to determine their challenges.

The same kind of attack applies obviously also to the issuing protocols for the other Fiat-Shamir type secret-key certificates, since they are all derived according to the same technique. Since the modification for the Schnorr-based scheme has been mentioned in the description of [4], for the sake of concreteness the attack for the Schnorr-based issuing protocol will now be described in detail.

In the restrictive blinding issuing protocol for the Schnorr secret-key certificate scheme, using notation in line with the above description, signer $\mathcal{S}$ uses a public key $(\mathrm{desc}(G_q), g, h, g_1, \mathcal{H}(\cdot))$ and a corresponding secret key $(\log_g h, \log_g g_1)$, denoted by (x, y). Here, $\mathrm{desc}(G_q)$ denotes the polynomial-size description (including the specification of q) of a group G_q of prime order q for which polynomial-time algorithms are known to multiply, determine equality of elements, test membership, and to randomly select elements, and for which no feasible algorithms for computing discrete logarithms are known. Furthermore, g, h and g_1 are elements of G_q, and $\mathcal{H}(\cdot)$ is a polynomial-size description of a correlation-free one-way hash-function that maps its inputs to $\mathbb{Z}_{2^t}$ for some appropriate t. A *secret-key certificate* of $\mathcal{S}$ on a public key h_i of receiver $\mathcal{R}_i$ is a pair $(r, c) \in G_q \times \mathbb{Z}_{2^t}$ such that

$$c = \mathcal{H}(h_i, g^r (h\, h_i)^{-c}).$$

A secret key of $\mathcal{R}_i$ corresponding to its public key h_i is a pair $(s_{0i}, s_{1i}) \in \mathbb{Z}_q \times \mathbb{Z}_q$ such that

$$h_i = g_1^{s_{0i}} g^{s_{1i}}.$$

In one execution of the issuing protocol $\mathcal{R}_i$ receives a certified key pair (s_{0i}, s_{1i}), h_i', (r', c'), with the blinding-invariant predicate of the secret key being equal to $s_{0i} \bmod q$. The number $g_1^{s_{0i}}$ will be denoted by h_i. The issuing protocol is as follows:

Step 1. $\mathcal{S}$ generates at random a number $w \in \mathbb{Z}_q$, and sends $a := g^w$ to $\mathcal{R}_i$.

Step 2. $\mathcal{R}_i$ generates at random three numbers $s_{1i}, t_1, t_2 \in \mathbb{Z}_q$. $\mathcal{R}_i$ computes $h_i' := h_i g^{s_{1i}}$, $c' := \mathcal{H}(h_i', g^{t_1}(h\, h_i)^{t_2} a)$, and sends $c := c' + t_2 \bmod q$ to $\mathcal{S}$.

Step 3. $\mathcal{S}$ sends $r := c(x + y s_{0i}) + w \bmod q$ to $\mathcal{R}_i$.

$\mathcal{R}_i$ accepts if and only if $g^r (h\, h_i)^{-c} = a$. If this verification holds, $\mathcal{R}_i$ computes $r' := r + t_1 + c' s_{1i} \bmod q$.

Let s_{0i} denote the blinding-invariant number for $\mathcal{R}_i$, and $s_{0j} \neq s_{0i} \bmod q$ that for $\mathcal{R}_j$; the corresponding "not-yet-blinded" public keys are h_i and h_j. The alternative attack on two parallel executions of this issuing protocol is the following:

(Step 1 for $\mathcal{R}_i$) $\mathcal{S}$ generates at random a number $w_i \in \mathbb{Z}_q$, and sends $a_i := g^{w_i}$ to $\mathcal{R}_i$.

(Step 1 for $\mathcal{R}_j$) $\mathcal{S}$ generates at random a number $w_j \in \mathbb{Z}_q$, and sends $a_j := g^{w_j}$ to $\mathcal{R}_j$.

(Cooperation between $\mathcal{R}_i$ and $\mathcal{R}_j$) $\mathcal{R}_i$ and $\mathcal{R}_j$ decide on an arbitrary number $\alpha \in \mathbb{N}$, and compute $h_k := h_i^{\alpha} h_j^{1-\alpha}$. They then compute $c_k := \mathcal{H}(h_k, a_i a_j)$.

(Step 2 for $\mathcal{R}_i$) $\mathcal{R}_i$ sends $c_i := \alpha c_k \bmod q$ to $\mathcal{S}$.

(Step 2 for $\mathcal{R}_j$) $\mathcal{R}_j$ sends $c_j := (1-\alpha) c_k \bmod q$ to $\mathcal{S}$.

(Step 3 for $\mathcal{R}_i$) $\mathcal{S}$ sends $r_i := c_i(x + y s_{0i}) + w_i \bmod q$ to $\mathcal{R}_i$.

(Step 3 for $\mathcal{R}_j$) $\mathcal{S}$ sends $r_j := c_j(x + y s_{0j}) + w_j \bmod q$ to $\mathcal{R}_j$.

$\mathcal{R}_i$ and $\mathcal{R}_j$ accept if and only if

$$g^{r_i}(h\, h_i)^{-c_i} = a_i \quad \text{and} \quad g^{r_j}(h\, h_j)^{-c_j} = a_j.$$

If this verification holds, then $\mathcal{R}_i$ and $\mathcal{R}_j$ compute $r_k := r_i r_j$.

Proposition 2 *Let s_{0k} denote $\alpha s_{0i} + (1-\alpha) s_{0j} \bmod q$. If $\mathcal{R}_i$ and $\mathcal{R}_j$ accept, then*

$$(s_{0k}, 1),\ h_k,\ (r_k, c_k)$$

is a certified key pair.

3. Conclusion.

If using the secret-key certificate issuing protocols described in [3] only in sequential mode is insufficient for some highly demanding practical application, then obviously one must design different issuing protocols for these secret-key certificates that are secure even in parallel mode (or design new secret-key certificate schemes altogether). Although the modification technique proposed in [3] has been shown in this note to be ineffective, there may very well exist other modifications that are secure in parallel mode.

It is clear that the main problem stems from the multiplicative relation between a_i and a_j when algebraically combining the respective verification relations for $\mathcal{R}_i$ and $\mathcal{R}_j$. One trivial attempt to destroy this multiplicative relation would be to let $\mathcal{S}$ send $f(a)$ instead of a in the first move of the issuing protocols, where $f(\cdot)$ is a correlation-free one-way (hash-)function or permutation. This still enables $\mathcal{R}_i$ to verify the correctness of the response r in the third move (by verifying whether $f(r^v(hh_i)^{-c})$ resp. $f(g^r(hh_i)^{-c})$ is equal to the number sent by $\mathcal{S}$ in the first move). Since it then is infeasible to determine a number a_k and constants l_i and l_j such that $f(a_i)^{l_i} f(a_j)^{l_j} = f(a_k)$, parallel attacks such as those described above and in Section 3.6 of [3] will not work. There is one problem with this modification, though: $\mathcal{R}_i$ can no longer blind the certified public keys (in particular, $f(a)$) that it retrieves in the modified issuing protocol, and so this particular modification fails on another account.

Note that the restrictive blind issuing protocol for Schnorr-based *public-key* certificates, developed in [1] (see also [5]), seems resistant to parallel mode attacks such as those described in [3] and in this note, and so for highly demanding Discrete Log based systems one can always use that particular certificate scheme. A modification technique for secret-key certificate issuing protocols that does seem to offer security in parallel mode is the subject of forthcoming work.

REFERENCES

1. Brands, S., "Untraceable Off-Line Cash in Wallet with Observers," Advances in Cryptology – CRYPTO '93, Lecture Notes in Computer Science, no. 773, Springer-Verlag, pp. 302–318. An pre-print appeared as: "An efficient off-line electronic cash system based on the representation problem," Centrum voor Wiskunde en Informatica (CWI), Report CS-R9323, March 1993. Available by anonymous ftp from: ftp.cwi.nl:/pub/CWIreports/AA/CS-R9323.ps.Z.

2. Brands, S., manuscript (1993) part (i): "Secret-Key Certificates," Centrum voor Wiskunde en Informatica (CWI), Report CS-R9510, Februari 1995. Available by anonymous ftp from: ftp.cwi.nl:/pub/CWIreports/AA/CS-R9510.ps.Z.

3. Brands, S., manuscript (1993) part (ii): "Restrictive Blinding of Secret-Key Certificates (extended abstract)," Advances in Cryptology – EUROCRYPT '95, Lecture Notes in Computer Science, Springer-Verlag. See for full paper: Centrum voor Wiskunde en Informatica (CWI), Report CS-R9509, Februari 1995. Available by anonymous ftp from: ftp.cwi.nl:/pub/CWIreports/AA/CS-R9509.ps.Z.

4. Brands, S., manuscript (1993) part (iii): "Off-Line Electronic Cash Based on

Secret-Key Certificates," Proc. of the Second International Symposium of Latin American Theoretical Informatics (LATIN '95), Valparaíso, Chili, April 3–7, 1995. See also: Centrum voor Wiskunde en Informatica (CWI), Report CS-R9506, Januari 1995. Available by anonymous ftp from: ftp.cwi.nl:/pub/CWIreports/AA/CS-R9506.ps.Z.

5. Brands, S., "Electronic Cash on the Internet," Proc. of the Internet Society 1995 Symposium on Network and Distributed System Security, San Diego, California, Februari 16-17, 1995.

6. Brickell, E., McCurley, K., "An Interactive Identification Scheme Based on Discrete Logarithms and Factoring," Journal of Cryptology, Vol. 5, No. 1 (1992), pp. 29–39.

7. Feige, U., Fiat, A., Shamir, A., "Zero-Knowledge Proofs of Identity," Journal of Cryptology, Vol. 1, No. 2 (1988), pp. 77–94.

8. Fiat, A. and Shamir, A., "How to prove yourself: practical solutions to identification and signature problems," Advances in Cryptology – CRYPTO '86, Lecture Notes in Computer Science, Springer-Verlag, pp. 186-194.

9. Guillou, L., Quisquater, J.-J., "A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory," Advances in Cryptology – EUROCRYPT '88, Lecture Notes in Computer Science, no. 330, Springer-Verlag, pp. 123-128.

10. Okamoto, T., "Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes," Advances in Cryptology – CRYPTO '92, Lecture Notes in Computer Science, no. 740, Springer-Verlag, pp. 31–53.

11. Okamoto, T., Ohta, K., "Divertible Zero-Knowledge Interactive Proofs and Commutative Random Self-Reducibility," Advances in Cryptology – EUROCRYPT '89, Lecture Notes in Computer Science, no. 434, Springer-Verlag, pp. 481–496.

12. Schnorr, C, "Efficient Signature Generation by Smart Cards," Journal of Cryptology, Vol. 4, No. 3 (1991), pp. 161-174.