Fair Blind Signatures without Random Oracles

Georg Fuchsbauer and Damien Vergnaud

École normale supérieure, LIENS - CNRS - INRIA, Paris, France
http://www.di.ens.fr/{~fuchsbau,~vergnaud}

Abstract. A fair blind signature is a blind signature with revocable anonymity and unlinkability, i.e. an authority can link an issuing session to the resulting signature and trace a signature to the user who requested it. In this paper we first revisit the security model for fair blind signatures given by Hufschmitt and Traoré in 2007. We then give the first practical fair blind signature scheme with a security proof in the standard model. Our scheme satisfies a stronger variant of the Hufschmitt-Traoré model.

Keywords: Blind signatures, revocable anonymity, standard model, Groth-Sahai proof system.

1 Introduction

A blind signature scheme is a protocol for obtaining a signature from an issuer (signer) such that the issuer's view of the protocol cannot be linked to the resulting message/signature pair. Blind signatures are employed in privacy-related protocols where the issuer and the message author are different parties (e.g., e-voting or e-cash systems). However, blind signature schemes provide perfect unlinkability and could therefore be misused by dishonest users. Fair blind signatures were introduced by Stadler, Piveteau and Camenisch [SPC95] to prevent abuse of unlinkability. They allow two types of blindness revocation: linking a signature to the user who asked for it and identifying a signature that resulted from a given signing session. A security model for fair blind signatures was introduced by Hufschmitt and Traoré [HT07].

We first revisit this security model and propose a stronger variant. We then present the first efficient fair blind signature scheme with a standard-model security proof (i.e. without resorting to the random-oracle heuristic) in the strengthened model. We make extensive use of the non-interactive proof system due to Groth and Sahai [GS08] and of the *automorphic signatures* recently introduced by Fuchsbauer [Fuc09]; we do not rely on interactive assumptions. We note that this is an extended abstract and refer to the full version [FV10] for detailed proofs and an improved scheme based on recent results in [Fuc09].

1.1 Prior Work

The concept of *blind signatures* was introduced by Chaum in [Cha83]. A blind signature scheme is a cryptographic primitive that allows a user to obtain from

D.J. Bernstein and T. Lange (Eds.): AFRICACRYPT 2010, LNCS 6055, pp. 16–33, 2010.

the *issuer* (signer) a digital signature on a message of the user's choice in such a way that the issuer's view of the protocol cannot be linked to the resulting message/signature pair. Blind signatures have numerous applications including e-cash: they prevent linking withdrawals and payments made by the same customer. However, the impossibility of this linking might lead to frauds (money laundering, blackmailing, ...); some applications therefore require means to identify the resulting signature from the transcript of a signature-issuing protocol or to link a message/signature pair to user who requested it.

Fair blind signatures were introduced by Stadler, Piveteau and Camenisch in [SPC95] to provide these means. Several schemes have been proposed since then [SPC95, AO01, HT07] with applications to e-cash [GT03] or e-voting [CGT06]. In [HT07], Hufschmitt and Traoré presented the first formal security model for fair blind signatures and a scheme based on bilinear maps satisfying it in the random oracle model under an interactive assumption. In a recent independent work, Rückert and Schröder [RS10] proposed a *generic* construction of fair *partially* blind signatures [AF96].

1.2 Our Contribution

As a first contribution, we strengthen the security model proposed in [HT07]. In our model, opening a transcript of an issuing session not only reveals information to identify the resulting signature, but also the user that requested it.

We give a definition of blindness analogously to [Oka06], but additionally provide tracing oracles to the adversary; in contrast to [HT07], this models *active* adversaries. We propose a traceability notion that implies the original one. Finally, we formalize the non-frameability notions analogously to [BSZ05], where it is the adversary's task to output a framing signature (or transcript) *and a proof.* (In [HT07] the experiment produces the proof, limiting thus the adversary.) We believe that our version of signature non-frameability is more intuitive: no corrupt issuer can output a transcript, an opening framing a user, and a proof. (In [HT07] the adversary must output a message/signature pair such that an honest transcript opens to it.) (See §2.3 for the details.)

In 2008, Groth and Sahai [GS08] proposed a way to produce efficient non-interactive zero-knowledge (NIZK) and non-interactive witness-indistinguishable (NIWI) proofs for (algebraic) statements related to groups equipped with a bilinear map. In particular, they give proofs of satisfiability of *pairing-product equations* (cf. §4.2 and [BFI$^+$10] for efficiency improvements for proof verification). In [Fuc09], Fuchsbauer introduced the notion of *automorphic signatures* whose verification keys lie in the message space, messages and signatures consist of group elements only, and verification is done by evaluating a set of pairing-product equations (cf. §5). Among several applications, he constructed an (automorphic) blind signature in the following way: the user commits to the message, and gives the issuer a randomized message; the issuer produces a "pre-signature" from which the user takes away the randomness to recover a signature. The actual signature is then a Groth-Sahai NIWI proof of knowledge of a signature, which guarantees unlinkability to the issuing.

In this paper, we modify Fuchsbauer's blind signature scheme in order to construct the first practical fair blind signature scheme with a security reduction in the standard model. Our security analysis does not introduce any new computational assumptions and relies only on falsifiable assumptions [Nao03] (cf. §3). First, we extend Fuchsbauer's automorphic signature so it can sign three messages at once. Then, to achieve blindness even against adversaries provided with tracing oracles, we use Groth's technique from [Gro07] to achieve CCA-anonymous group signatures: instead of just committing to the tracing information, we additionally encrypt it (using Kiltz' tag-based encryption scheme [Kil06]) and provide NIZK proofs of consistency with the commitments. In order to achieve the strengthened notion of non-frameability, we construct simulation-sound NIZK proofs of knowledge of a Diffie-Hellman solution which consist of group elements only and are verified by checking a set of pairing-product equations (i.e. they are Groth-Sahai compatible).

Since messages and signatures consist of group elements only and their verification predicate is a conjunction of pairing-product equations, our fair blind signatures are Groth-Sahai compatible themselves which makes them perfectly suitable to design efficient fair e-cash systems following the approach proposed in [GT03]. In addition, our scheme is compatible with the "generic" variant[1] of Votopia [OMA$^+$99] proposed by Canard, Gaud and Traoré in [CGT06]. Combined with a suitable mix-net (e.g. [GL07]), it provides a practical electronic voting protocol in the standard model including public verifiability, and compares favorably with other similar systems in terms of computational cost.

2 The Model

2.1 Syntax

Definition 1. *A* fair blind signature scheme *is a 10-tuple*

$$(\mathsf{Setup}, \mathsf{IKGen}, \mathsf{UKGen}, \mathsf{Sign}, \mathsf{User}, \mathsf{Ver}, \mathsf{TrSig}, \mathsf{TrId}, \mathsf{ChkSig}, \mathsf{ChkId})$$

of (interactive) (probabilistic) polynomial-time Turing machines ((P)PTs):

Setup *is a PPT that takes as input an integer* λ *and outputs the* parameters pp *and the* revocation key rk*. We call* λ *the* security parameter.

IKGen *is a PPT that takes as input the parameters* pp *and outputs a pair* (ipk, isk)*, the* issuer's public and secret key.

UKGen *is a PPT that takes as input the parameters* pp *and outputs a pair* (upk, usk)*, the* user's public and secret key.

Sign *and* User *are interactive PPTs such that* User *takes as inputs* pp*, the issuer's public key* ipk*, the user's secret key* usk *and a bit string* m*;* Sign *takes as input* pp*, the issuer's secret key* isk *and user public key* upk*.* Sign *and* User *engage in the* signature-issuing protocol *and when they stop,* Sign *outputs* `completed` *or* `not-completed` *while* User *outputs* $\perp$ *or a bit string* σ*.*

[1] This variant was used during the French referendum on the European Constitution in May 2005.

Ver *is a deterministic PT (DPT) that on input the parameters* pp*, an issuer public key* ipk *and a pair of bit strings* (m, σ) *outputs either* 0 *or* 1*. If it outputs* 1 *then* σ *is a* valid signature *on the* message m*.*

TrSig *is a DPT that on input* pp*, an issuer public key* ipk*, a transcript* $trans$ *of a signature-issuing protocol and a revocation key* rk *outputs three bit strings* (upk, id_σ, π)*.*

TrId *is a DPT that on input* pp*, an issuer public key* ipk*, a message/signature pair* (m, σ) *for* ipk *and a revocation key* rk *outputs two bit strings* (upk, π)*.*

ChkSig *is a DPT that on input* pp*, an issuer public key* ipk*, a transcript of a signature issuing protocol, a pair message/signature* (m, σ) *for* ipk *and three bit strings* (upk, id_σ, π)*, outputs either* 0 *or* 1*.*

ChkId *is a DPT that on input* pp*, an issuer public key* ipk*, a message/signature pair* (m, σ) *for* ipk *and two bit strings* (upk, π)*, outputs either* 0 *or* 1*.*

For all $\lambda \in \mathbb{N}$*, all pairs* (pp, rk) *output by* $\mathsf{Setup}(\lambda)$ *all pairs* (ipk, isk) *output by* $\mathsf{IKGen}(pp)$*, and all pairs* (upk, usk) *output by* $\mathsf{UKGen}(pp)$*:*

1. *if* Sign *and* User *follow the signature-issuing protocol with input* (pp, isk, upk) *and* (pp, usk, ipk, m) *respectively, then* Sign *outputs* `completed` *and* User *outputs a bit string* σ *that satisfies* $\mathsf{Ver}(ipk, (m, \sigma)) = 1$*;*
2. *on input* ipk*, the transcript* $trans$ *of the protocol and* rk*,* TrSig *outputs three bit strings* (upk, id_σ, π) *s.t.* $\mathsf{ChkSig}(pp, ipk, trans, (m, \sigma), (upk, id_\sigma, \pi)) = 1$*;*
3. *on input* ipk*, the pair* (m, σ) *and* rk*,* TrId *outputs two bit strings* (upk, π) *such that* $\mathsf{ChkId}(pp, ipk, (m, \sigma), (upk, \pi)) = 1$*.*

2.2 Security Definitions

To define the security notions for fair blind signatures, we use a notation similar to the one in [BSZ05] used in [HT07]:

HU denotes the set of honest users and *CU* is the set of corrupted users.

AddU is an *add-user* oracle. The oracle runs $(upk, usk) \leftarrow \mathsf{UKGen}(pp)$, adds upk to *HU* and returns it to the adversary.

CrptU is a *corrupt-user* oracle. The adversary calls it with a pair (upk, usk) and upk is added to the set *CU*.

USK is a *user-secret-key* oracle enabling the adversary to obtain the private key usk for some $upk \in HU$. The oracle transfers upk to *CU* and returns usk.

User is an *honest-user* oracle. The adversary impersonating a corrupt issuer calls it with (upk, m). If $upk \in HU$, the experiment simulates the honest user holding upk running the signature issuing protocol with the adversary for message m. If the issuing protocol completed successfully, the adversary is given the resulting signature. The experiment keeps a list *Set* with entries of the form $(upk, m, trans, \sigma)$, to record an execution of User, where $trans$ is the transcript of the issuing session and σ is the resulting signature. (Note that only valid σ's (i.e. the protocol was successful) are written to *Set*.

Sign is a *signing* oracle. The adversary impersonating a corrupt user can use it to run the issuing protocol with the honest issuer. The experiment keeps a list *Trans* in which the transcripts $trans_i$ resulting from Sign calls are stored.

$\mathsf{Challenge}_b$ is a *challenge* oracle, which (w.l.o.g.) can only be called once. The adversary provides two user public keys upk_0 and upk_1 and two messages m_0 and m_1. The oracle first simulates User on inputs (pp, ipk, usk_b, m_b) and then, in a second protocol run, simulates User on inputs $(pp, ipk, usk_{1-b}, m_{1-b})$. Finally, the oracle returns (σ_0, σ_1), the resulting signatures on m_0 and m_1.

TrSig (resp. TrId) is a *signature (resp. identity) tracing* oracle. When queried on the transcripts (or messages) emanating from a $\mathsf{Challenge}$ call, they return $\perp$.

Figure 1 formalizes the experiments for the following security notions:

Blindness. Not even the issuer with access to tracing oracles can link a message/signature pair to the signature-issuing session it stems from.

Identity Traceability. No coalition of users can produce a set of signatures containing signatures which cannot be linked to an identity.

Signature Traceability. No coalition of users is be able to produce a message/signature pair which is not traced by any issuing transcript or two pairs which are traced by the same transcript.

Identity Non-Frameability. No coalition of issuer, users and tracing authority should be able to provide a signature and a proof that the signature opens to an honest user who did not ask for the signature.

Signature Non-Frameability. No coalition of issuer, users and tracing authority should be able to provide a transcript that either wrongfully opens to an honest signature or an honest user.

We say that a fair blind signature achieves *blindness* if for all p.p.t. adversaries $\mathcal{A}$, the following is negligible: $| \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{blind-1}} = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{blind-0}} = 1] - \frac{1}{2}$. The remaining security notions are achieved if for all p.p.t. $\mathcal{A}$, the probability that the corresponding experiment returns 1 is negligible.

2.3 A Note on the Hufschmitt-Traoré Security Notions

Blindness. In [HT07], the challenge oracle (called “Choose”) is defined as follows: the adversary provides two user public keys upk_0 and upk_1 and a message, and obtains a signature under upk_b. This gives a weak security guarantee, as the adversary—who should impersonate the issuer—cannot actively participate in the issuing of the challenge signature. We define our oracle in the spirit of [Oka06]: the adversary chooses two users (and messages) which interact with him in random order; he gets to see both resulting signatures and has to determine the order of issuing.

Traceability Notions. Intuitively, identity traceability means that no coalition of users and the authority can create a message/signature pair that is not traceable to a user, which is what was formalized in [HT07].

We propose the following experiment leading to a stronger notion: the adversary gets the authority’s key and impersonates corrupt users, who, via the Sign oracle can request signatures from the honest issuer. The latter is simulated by the experiment and keeps a set *Trans* of transcripts of oracle calls. Eventually,

```
$\mathbf{Exp}_{\mathcal{A}}^{\text{blind-b}}(\lambda)$
    $(pp, rk) \leftarrow \mathsf{Setup}(1^\lambda)$; $(ipk, isk) \leftarrow \mathsf{IKGen}(pp)$
    $b' \leftarrow \mathcal{A}(pp, ipk, isk : \mathsf{AddU}, \mathsf{CrptU}, \mathsf{USK}, \mathsf{Challenge}_b, \mathsf{User}, \mathsf{TrSig}, \mathsf{TrId})$
    return $b'$

$\mathbf{Exp}_{\mathcal{A}}^{\text{IdTrac}}(\lambda)$
    $(pp, rk) \leftarrow \mathsf{Setup}(1^\lambda)$; $(ipk, isk) \leftarrow \mathsf{IKGen}(pp)$; $Trans \leftarrow \emptyset$
    $(m_1, \sigma_1, \ldots, m_n, \sigma_n) \leftarrow \mathcal{A}(pp, ipk, rk : \mathsf{AddU}, \mathsf{CrptU}, \mathsf{USK}, \mathsf{Sign})$
    for $i = 1 \ldots |Trans|$ do $(upk_i, id_i, \pi_i) \leftarrow \mathsf{TrSig}(pp, rk, ipk, trans_i)$
    for $i = 1 \ldots n$ do $(upk'_i, \pi'_i) \leftarrow \mathsf{TrId}(pp, rk, ipk, m_i, \sigma_i)$
    if $\exists\, i : upk'_i = \bot$ or $\mathsf{ChkId}(pp, ipk, (m_i, \sigma_i), upk'_i, \pi'_i) = 0$ then return 1
    if some $upk$ appears more often in $(upk'_1, \ldots, upk'_n)$ than in
                                              $(upk_1, \ldots, upk_{|Trans|})$ then return 1;  else return 0

$\mathbf{Exp}_{\mathcal{A}}^{\text{IdNF}}(\lambda)$
    $(pp, rk) \leftarrow \mathsf{Setup}(1^\lambda)$; $(ipk, isk) \leftarrow \mathsf{IKGen}(pp)$
    $Set \leftarrow \emptyset$; $HU \leftarrow \emptyset$; $CU \leftarrow \emptyset$
    $(upk, m, \sigma, \pi) \leftarrow \mathcal{A}(pp, ipk, isk, rk : \mathsf{AddU}, \mathsf{CrptU}, \mathsf{USK}, \mathsf{User})$
    if $\mathsf{Ver}(pp, ipk, m, \sigma) = 0$ or $\mathsf{ChkId}(pp, ipk, m, \sigma, upk, \pi) = 0$ then return 0
    if $(upk, m, \cdot, \sigma) \notin Set$ and $upk \in HU$ then return 1; else return 0

$\mathbf{Exp}_{\mathcal{A}}^{\text{SigTrac}}(\lambda)$
    $(pp, rk) \leftarrow \mathsf{Setup}(1^\lambda)$; $(ipk, isk) \leftarrow \mathsf{IKGen}(pp)$; $Trans \leftarrow \emptyset$
    $(m_1, \sigma_1, m_2, \sigma_2) \leftarrow \mathcal{A}(pp, ipk, rk : \mathsf{AddU}, \mathsf{CrptU}, \mathsf{USK}, \mathsf{Sign})$
    let $Trans = (trans_i)_{i=1}^n$; for $i = 1 \ldots n$ do $(upk_i, id_i, \pi_i) \leftarrow \mathsf{TrSig}(pp, rk, ipk, trans_i)$
    if $\mathsf{Ver}(pp, ipk, m_1, \sigma_1) = 1$ and
                          $\forall\, i : \mathsf{ChkSig}(pp, ipk, trans_i, m_1, \sigma_1, upk_i, id_i, \pi_i) = 0$ then return 1
    if $(m_1, \sigma_1) \neq (m_2, \sigma_2)$ and $\mathsf{Ver}(pp, ipk, m_1, \sigma_1) = 1$ and $\mathsf{Ver}(pp, ipk, m_2, \sigma_2) = 1$
                          and $\exists\, i : \mathsf{ChkSig}(pp, ipk, trans_i, m_1, \sigma_1, upk_i, id_i, \pi_i) =$
                              $= \mathsf{ChkSig}(pp, ipk, trans_i, m_2, \sigma_2, upk_i, id_i, \pi_i)) = 1$
      then return 1; else return 0

$\mathbf{Exp}_{\mathcal{A}}^{\text{SigNF}}(\lambda)$
    $(pp, rk) \leftarrow \mathsf{Setup}(1^\lambda)$; $(ipk, isk) \leftarrow \mathsf{IKGen}(pp)$
    $Set \leftarrow \emptyset$; $HU \leftarrow \emptyset$; $CU \leftarrow \emptyset$
    $(trans^*, m^*, \sigma^*, upk^*, id^*_\sigma, \pi^*) \leftarrow \mathcal{A}(pp, ipk, isk, rk : \mathsf{AddU}, \mathsf{CrptU}, \mathsf{USK}, \mathsf{User})$
    let $Set = (upk_i, m_i, trans_i, \sigma_i)_{i=1}^n$
    if $\exists\, i : trans^* \neq trans_i$ and $\mathsf{ChkSig}(pp, ipk, trans^*, m_i, \sigma_i, upk^*, id^*_\sigma, \pi^*) = 1$
      then return 1
    if $(\forall\, i : upk^* = upk_i \Rightarrow trans^* \neq trans_i\,)$
                              and $\mathsf{ChkSig}(\ldots, trans^*, m^*, \sigma^*, upk^*, id^*_\sigma, \pi^*) = 1$
      then return 1; else return 0
```

Fig. 1. Security experiments for fair blind signatures

the adversary outputs a set of message/signature pairs. The experiment opens all transcripts to get a list of users to which signatures were issued. Another list of users is constructed by opening the returned signatures. The adversary wins if there exists a user who appears more often in the second list than in the first, or if $\perp$ is in the second list, or if any of the proofs output by the opening algorithm do not verify. Note that the notion of [HT07] is implied by ours.

Non-Frameability Notions. Non-frameability means that not even a coalition of everyone else can "frame" an honest user. For example, no adversary can output a signature which opens to a user who did not participate in its issuing. In [HT07], the adversary outputs a message/signature pair, which is then opened by the experiment to determine if it "framed" a user. Analogously to [BSZ05] (who defined non-frameability for group signatures), we define a strictly stronger notion requiring the adversary to output an incriminating signature, an honest user, *and a valid proof* that the signature opens to that user. Note that only this formalization makes the π output by the tracing algorithms a proof, as it guarantees that no adversary can produce a proof that verifies for a false opening.

Identity Non-Frameability. In [HT07], the adversary wins if it produces a pair (m, σ) such that, when opened to *upk*, we have $(m, \sigma, \mathit{upk}) \notin \mathit{Set}$. This seems to guarantee *strong* unforgeability where an adversary modifying a signature returned by the experiment wins the game. This is however not the case in the scheme proposed in [HT07]: the final signature is a proof of knowledge of some values computed by the issuer made non-interactive by the Fiat-Shamir heuristic; hence from a given signature issuing session the user may derive several valid signatures on a message m. For that reason, the model in [HT07] considers two signatures different only if the underlying secrets are different. We adopt the same convention in this paper in that we consider two signatures equivalent if they have the same (public) *identifier*.

Signature Non-Frameability. Non-frameability of signature tracing intuitively means: even if everyone else colludes against an honest user, they cannot produce a transcript that opens to an honest signature. In the definition proposed in [HT07], the adversary plays the issuer in that he gets his secret key. However, he has no possibility to communicate with honest users since the *challenger* plays the issuer in the signature-issuing sessions with honest users and the adversary only gets the transcripts. His goal is to produce a *new* message/signature pair (one that does not emanate from a User-oracle call) such that an honest transcript opens to it.

We give the following security notion which we think is more intuitive. No corrupt issuer can produce a transcript of an issuing session and one of the following: either a public key of an honest user and a proof that this user participated in the transcript whereas she did not; or a signature identifier of an honest signature coming from a different session and a proof that the transcript opens to it. Similarly to signatures we consider two transcripts equivalent if the contain the same user randomness and the same issuer randomness.

Unforgeability. Consider an adversary that breaks the classical security notion for blind signatures, one-more unforgeability, i.e. after $q - 1$ Sign-oracle queries, he outputs q signatures on different messages. We show that the adversary must have broken signature traceability: indeed since there are more signatures than transcripts, either there is a signature which no transcripts points to, or there is a transcript that points to two signatures.

3 Assumptions

A (symmetric) *bilinear group* is a tuple $(p, \mathbb{G}, \mathbb{G}_T, e, G)$ where $(\mathbb{G}, \cdot)$ and $(\mathbb{G}_T, \cdot)$ are two cyclic groups of prime order p, G is a generator of $\mathbb{G}$, and $e \colon \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a non-degenerate bilinear map, i.e. $\forall\, U, V \in \mathbb{G}\ \forall\, a, b \in \mathbb{Z}$: $e(U^a, V^b) = e(U, V)^{ab}$, and $e(G, G)$ is a generator of $\mathbb{G}_T$.

The *Decision Linear (DLIN) Assumption* [BBS04], in $(p, \mathbb{G}, \mathbb{G}_T, e, G)$ states that given $(G^\alpha, G^\beta, G^{r\alpha}, G^{s\beta}, G^t)$ for random $\alpha, \beta, r, s \in \mathbb{Z}_p$, it is hard to decide whether $t = r + s$ or t is random.

The following two assumptions were introduced by [FPV09] and [Fuc09], respectively. Under the *knowledge of exponent assumption* [Dam92], the first is equivalent to SDH [BB04] and the second is equivalent to computing discrete logarithms.

Assumption 1 (q-DHSDH). *Given $(G, H, K, X = G^x) \in \mathbb{G}^4$ and $q - 1$ tuples*

$$\big(A_i = (KG^{v_i})^{\frac{1}{x+d_i}},\ C_i = G^{d_i},\ D_i = H^{d_i},\ V_i = G^{v_i},\ W_i = H^{v_i}\big)_{i=1}^{q-1}\ ,$$

for $d_i, v_i \leftarrow \mathbb{Z}_p$, it is hard to output a new tuple $(A, C, D, V, W) \in \mathbb{G}^5$ satisfying

$$e(A, XC) = e(KV, G) \qquad e(C, H) = e(G, D) \qquad e(V, H) = e(G, W) \qquad (1)$$

The next assumption states that, given $(G, H, T) \in \mathbb{G}^3$, it is hard to produce a non-trivial (G^m, H^m, G^r, H^r) such that $G^m = T^r$.

Assumption 2 (HDL). *Given a random triple $(G, H, T) \in \mathbb{G}^3$, it is hard to output $(M, N, R, S) \neq (1, 1, 1, 1)$ such that*

$$e(R, T) = e(M, G) \qquad e(M, H) = e(G, N) \qquad e(R, H) = e(G, S) \qquad (2)$$

4 Tools

We recall some tools from the literature which we use to construct our scheme.

4.1 A Signature Scheme to Sign Group Elements

We present the signature scheme from [Fuc09], which is secure against chosen-message attacks under Assumptions 1 and 2. Its message space is the set of *Diffie-Hellman pairs* $\mathcal{DH} := \{(A, B) \in \mathbb{G}^2 \,|\, \exists \alpha : A = G^\alpha, B = H^\alpha\}$ w.r.t. two fixed generators $G, H \in \mathbb{G}$. Note that $(A, B) \in \mathcal{DH}$ iff $e(A, H) = e(G, B)$.

Scheme 1 (Sig$_1$).

Setup_1 Given $(p, \mathbb{G}, \mathbb{G}_T, e, G)$, choose additional generators $H, K, T \in \mathbb{G}$.

KeyGen_1 Choose $sk = x \leftarrow \mathbb{Z}_p$ and set $vk = G^x$.

Sign_1 To sign $(M, N) \in \mathcal{DH}$ with secret key x, choose $d, r \leftarrow \mathbb{Z}_p$ and output

$$\big(A := (KT^r M)^{\frac{1}{x+d}}, C := G^d, D := H^d, R := G^r, S := H^r\big) \ ,$$

Verify_1 (A, C, D, R, S) is valid on $(M, N) \in \mathcal{DH}$ under public key $vk = X$ iff

$$e(A, XC) = e(KM, G)\, e(T, R) \qquad \begin{array}{c} e(C, H) = e(G, D) \\ e(R, H) = e(G, S) \end{array} \tag{3}$$

4.2 Groth-Sahai Proofs

We sketch the results of Groth and Sahai [GS08] on proofs of satisfiability of sets of equations over a bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e, G)$. Due to the complexity of their methodology, we present what is needed for our results and refer to the full version of [GS08] for any additional details.

We define a key for *linear commitments*. Choose $\alpha, \beta, r_1, r_2 \leftarrow \mathbb{Z}_p$ and define $U = G^\alpha$, $V = G^\beta$, $W_1 := U^{r_1}$, $W_2 := V^{r_2}$, and W_3 which is either

- soundness setting: $W_3 := G^{r_1+r_2}$ (which makes $\vec{\mathbf{u}}$ a binding key); or
- witness-indistinguishable setting: $W_3 := G^{r_1+r_2-1}$ (making $\vec{\mathbf{u}}$ a hiding key)

Under key $ck = (U, V, W_1, W_2, W_3)$, a commitment to a group element $X \in \mathbb{G}$ using randomness $(s_1, s_2, s_3) \leftarrow \mathbb{Z}_p^3$ is defined as

$$\mathsf{Com}\big(ck, X; (s_1, s_2, s_3)\big) := \big(U^{s_1} W_1^{s_3},\ V^{s_2} W_2^{s_3},\ X G^{s_1+s_2} W_3^{s_3}\big) \ .$$

In the soundness setting, given the *extraction key* $ek := (\alpha, \beta)$, the committed value can be extracted from a commitment $\mathbf{c} = (c_1, c_2, c_3)$. On the other hand, in the witness-indistinguishable (WI) setting, $\mathbf{c}$ is equally distributed for every X. The two settings are indistinguishable under the DLIN assumption.

A *pairing-product equation* is an equation for variables $\mathcal{Y}_1, \ldots, \mathcal{Y}_n \in \mathbb{G}$ of the form

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{Y}_i) \prod_{i=1}^{n} \prod_{j=1}^{n} e(\mathcal{Y}_i, \mathcal{Y}_j)^{\gamma_{i,j}} \ = \ t_T \ ,$$

with $\mathcal{A}_i \in \mathbb{G}$, $\gamma_{i,j} \in \mathbb{Z}_p$ and $t_T \in \mathbb{G}_T$ for $1 \le i, j \le n$.

To prove satisfiability of a set of equations of this form, one first makes commitments to a satisfying witness (i.e. an assignment to the variables of each equation) and then adds a "proof" per equation. Groth and Sahai describe how to construct these: they are in $\mathbb{G}^{3\times 3}$ (or $\mathbb{G}^3$ when all $\gamma_{i,j} = 0$). In the soundness setting, if the proof is valid then Extr extracts the witness satisfying the equations. In the WI setting, commitments and proofs of different witnesses which both satisfy the same pairing-product equation are equally distributed.

4.3 Commit and Encrypt

In order to build CCA-anonymous group signatures, Groth [Gro07] uses the following technique: a group signature consists of linear commitments to a certified signature and Groth-Sahai proofs that the committed values constitute a valid signature. CPA-anonymity follows from WI of GS proofs: once the commitment key has been replaced by a perfectly hiding one, a group signature reveals no information about the signer. However, in order to simulate opening queries in the WI setting, some commitments are doubled with a *tag-based encryption* under Kiltz' scheme [Kil06] and a Groth-Sahai NIZK proof that the committed and the encrypted value are the same. To produce a group signature, the user first chooses a key pair for a one-time signature scheme, uses the verification key as the tag for the encryption and the secret key to sign the group signature.

By $\mathbf{Sig}_{\mathrm{ot}} = (\mathsf{KeyGen}_{\mathrm{ot}}, \mathsf{Sign}_{\mathrm{ot}}, \mathsf{Ver}_{\mathrm{ot}})$ we will denote the signature scheme discussed in §5.2 which satisfies the required security notion. By CEP (commit-encrypt-prove) we denote the following:

$$\begin{aligned}&\mathsf{CEP}(ck, pk, tag, msg;\ (\rho, r)) := \\ &\qquad \big(\mathsf{Com}(ck, msg;\ \rho),\ \mathsf{Enc}(pk, tag, msg;\ r),\ \mathsf{NizkEq}(ck, pk,\ tag;\ msg, \rho, r)\big)\end{aligned}$$

where Enc denotes Kiltz' encryption and NizkEq denotes a Groth-Sahai NIZK proof that the commitment and the encryption contain the same plaintext (cf. [Gro07]). We say that an output $\psi = (\mathbf{c}, C, \zeta)$ of CEP is *valid* if the ciphertext and the zero-knowledge proof are valid.

5 New Tools

5.1 A Scheme to Sign Three Diffie-Hellman Pairs

We extend the scheme from §4.1, so it signs three messages at once; we prove existential unforgeability (EUF) against adversaries making a particular chosen message attack (CMA): the first message is given (as usual) as a Diffie-Hellman pair, whereas the second and third message are queried as their logarithms; that is, instead of querying (G^v, H^v), the adversary has to give v explicitly. As we will see, this combines smoothly with our application.

Scheme 2 ($\mathbf{Sig}_3$).

$\mathsf{Setup}_3(\mathcal{G})$ Given $\mathcal{G} = (p, \mathbb{G}, \mathbb{G}_T, e, G)$, choose additional generators $H, K, T \in \mathbb{G}$.

$\mathsf{KeyGen}_3(\mathcal{G})$ Choose $sk = (x, \ell, u) \leftarrow \mathbb{Z}_p^3$ and set $vk = (G^x, G^\ell, G^u)$.

$\mathsf{Sign}_3((x, \ell, u), (M, N, Y, Z, V, W))$ A signature on $((M, N), (Y, Z), (V, W)) \in \mathcal{DH}^3$ under public key G^x, is defined as (for random $d, r \leftarrow \mathbb{Z}_p$)

$$\big(A := (KT^r M Y^\ell V^u)^{\frac{1}{x+d}}, C := G^d, D := H^d, R := G^r, S := H^r\big)$$

Verify_3 (A, C, D, R, S) is valid on messages $(M, N), (Y, Z), (V, W) \in \mathcal{DH}$ under a public key (X, L, U) iff

$$e(A, XC) = e(KM, G)\, e(T, R)\, e(L, Y)\, e(U, V) \qquad \begin{aligned} e(C, H) &= e(G, D) \\ e(R, H) &= e(G, S) \end{aligned} \tag{4}$$

Theorem 1. $\mathbf{Sig}_3$ *is existentially unforgeable against adversaries making chosen message attacks of the form* $((M_1, N_1), m_2, m_3)$.

Proof. Let (M_i, N_i, y_i, v_i) be the queries, $(A_i, C_i, D_i, R_i = G^{r_i}, S_i)$ be the responses. Let (M, N, Y, Z, V, W) and $(A, C, D, R = G^r, S)$ be a successful forgery. We distinguish 4 types of forgers (where $Y_i := G^{y_i}, V_i := G^{v_i}$):

Type I
$$\forall\, i : T^{r_i} M_i Y_i^{\ell} V_i^{u} \neq T^r M Y^{\ell} V^u \tag{5}$$

Type II
$$\exists\, i : T^{r_i} M_i Y_i^{\ell} V_i^{u} = T^r M Y^{\ell} V^u \;\wedge\; M_i Y_i^{\ell} V_i^{u} \neq M Y^{\ell} V^u \tag{6}$$

Type III
$$\exists\, i : M_i Y_i^{\ell} V_i^{u} = M Y^{\ell} V^u \;\wedge\; M_i V_i^{u} \neq M V^u \tag{7}$$

Type IV
$$\exists\, i : M_i Y_i^{\ell} V_i^{u} = M Y^{\ell} V^u \;\wedge\; M_i V_i^{u} = M V^u \tag{8}$$

Type I is reduced to DHSDH. Let $\big(G, H, K, (A_i, C_i, D_i, E_i, F_i)_{i=1}^{q-1}\big)$ be an instance. Choose and $t, \ell, u \leftarrow \mathbb{Z}_p$ and set $T = G^t$, $L = G^{\ell}$ and $U = G^u$. A signature on $(M_i, N_i, Y_i, Z_i, y_i, V_i, W_i, v_i)$ is (after a consistency check) answered as $(A_i, C_i, D_i, (E_i M_i^{-1} Y_i^{-\ell} V_i^{-u})^{1/t}, (F_i N_i^{-1} Z_i^{-\ell} W_i^{-u})^{1/t})$. After a successful forgery, return $(A, C, D, R^t M Y^{\ell} V^u, S^t N Z^{\ell} W^u)$, which is a valid DHSDH solution by (5).

Type II is reduced to HDL. Let (G, H, T) be an HDL instance. Generate the rest of the parameters and a public key and answer the queries by signing. After a successful forgery return the following, which is non-trivial by (6):

$$(M Y^{\ell} V^u M_i^{-1} Y_i^{-\ell} V_i^{-u}, N Z^{\ell} W^u N_i^{-1} Z_i^{-\ell} W_i^{-u}, R_i R^{-1}, S_i S^{-1}) \ .$$

Type III is reduced to HDL. Let (G, H, L) be an instance. Choose $K, T \leftarrow \mathbb{G}$ and $x, u \leftarrow \mathbb{Z}_p$ and return the parameters and public key $(X = G^x, L, U = G^u)$. Thanks to the y_i in the signing queries, we can simulate them: return $((K T^{r_i} M_i L^{y_i} V_i^u)^{\frac{1}{x+d_i}}, G^{d_i}, H^{d_i}, G^{r_i}, H^{r_i})$. We have $M V^u M_i^{-1} V_i^{-u} = Y_i^{\ell} Y^{-\ell} = L^{y_i - y}$ from (7), so from a successful forgery, we can return

$$(M V^u M_i^{-1} V_i^{-u}, N W^u N_i^{-1} W_i^{-u}, Y_i Y^{-1}, Z_i Z^{-1}) \ ,$$

which is non-trivial by (7).

Type IV is also reduced to HDL. Let (G, H, U) be an HDL instance. Choose $K, T \leftarrow \mathbb{G}$ and $x, \ell \leftarrow \mathbb{Z}_p$ and return the parameters and public key $(X = G^x, L = G^{\ell}, U)$. Thanks to the v_i in the signing queries, we can simulate them: return $((K T^{r_i} M_i Y_i^{\ell} U^{v_i})^{\frac{1}{x+d_i}}, G^{d_i}, H^{d_i}, G^{r_i}, H^{r_i})$. From a successful forgery of Type IV we have $M M_i^{-1} = U^{v_i - v}$ from (7), we can thus return $(M M_i^{-1}, N N_i^{-1}, V_i V^{-1}, W_i W^{-1})$, which is non-trivial, (M, N, Y, Z, V, W) being a valid forgery and $(Y, Z) = (Y_i, Z_i)$ by (8). □

5.2 A Simulation-Sound Non-interactive Zero-Knowledge Proof of Knowledge of a CDH Solution

Let (G, F, V) be elements of $\mathbb{G}$. We construct a simulation-sound non-interactive zero-knowledge (SSNIZK) proof of knowledge (PoK) of W s.t. $e(V, F) = e(G, W)$. We follow the overall approach by Groth [Gro06]. The common reference string (CRS) contains a CRS for Groth-Sahai (GS) proofs and a public key for a EUF-CMA signature scheme **Sig**. A proof is done as follows: choose a key pair for a one-time signature scheme $\mathbf{Sig}_{\mathrm{ot}}$, and make a witness-indistinguishable GS proof of the following: either to know W, a CDH solution for (G, F, V) or to know a signature on the chosen one-time key which is valid under the public key from the CRS;[2] finally sign the proof using the one-time key. A SSNIZKPoK is verified by checking the GS proofs and the one-time signature. Knowing the signing key corresponding to the key in the CRS, one can simulate proofs by using as a witness a signature on the one-time key.

We require that a proof consist of group elements only and is verified by checking a set of pairing-product equations. This can be achieved by using Scheme 1 and a one-time scheme to sign group elements using the commitment scheme in [Gro09] based on the DLIN assumption.[3]

6 A Fair Blind Signature Scheme

The basis of our protocol is the blind automorphic signature scheme from [Fuc09]: the user randomizes the message to be signed, the issuer produces a pre-signature from which the user obtains a signature by removing the randomness; the final signature is a Groth-Sahai (GS) proof of knowledge of the resulting signature.

In our scheme, in addition to the message, the issuer signs the user's public key, and an *identifier* of the signature, which the issuer and the user define jointly. Note that the issuer may neither learn the user's public key nor the identifier. To guarantee provable tracings, the user signs what she sends in the issuing protocol and the final signature. To prevent malicious issuers from producing a transcript that opens to an honest signature, the proof contains a SSNIZK proof of knowledge of the randomness introduced by the user. To achieve blindness against adversaries with tracing oracles, the elements that serve as proofs of correct tracing are additionally encrypted and the transcript (and final signature) is signed with a one-time key (cf. §4.3).

[2] [Gro06] shows how to express a disjunction of equation sets by a new set of equations.

[3] The strong one-time signature scheme from [Gro06] works as follows: the verification key is an (equivocable) Pedersen commitment to 0; to sign a message, the commitment is opened to the message using the trapdoor; putting a second trapdoor in the commitment scheme, we can simulate one signing query and use a forger to break the binding property of the commitment. In [Gro09], Groth proposes a scheme to commit to group elements which is computationally binding under DLIN. Using his scheme instead of Pedersen commitments, we can construct an efficient one-time signature on group elements s.t. signatures consist of group elements (see the full version [FV10] for the details).

To open a signature (i.e. to trace a user), the authority extracts tracing information from the commitments as well as signatures that act as proofs.

6.1 A Blind Signature Scheme

Setup. Choose a group $\mathcal{G} := (p, \mathbb{G}, \mathbb{G}_T, e, G)$ and parameters (H, K, T) for $\mathbf{Sig}_3$. Pick $F, H' \leftarrow \mathbb{G}$, a commitment and extraction key (ck, ek) for GS proofs, a key pair for tag-based encryption (epk, esk) and $sscrs$, a common reference string for SSNIZKPoK. Output $pp := (\mathcal{G}, G, H, K, T, F, H', ck, epk, sscrs)$ and $rk := ek$.

Key Generation. Both IKGen and UKGen are defined as KeyGen, i.e. the key generation algorithm for $\mathbf{Sig}_1$.

Signature Issuing. The common inputs are $(pp, ipk = G^x)$, the issuer's additional input is $isk = x$, the user's inputs are $(upk = G^y, usk = y,\ (M, N) \in \mathcal{DH})$.

User Choose $\eta, v' \leftarrow \mathbb{Z}_p$ and set $P = G^\eta, Q = F^\eta, V' = G^{v'}, W' = F^{v'}$.
Produce $\xi \leftarrow \mathsf{SSNIZKPoK}(sscrs, (P, V'), (Q, W'))$. [4]
Choose $(vk'_{\mathrm{ot}}, sk'_{\mathrm{ot}}) \leftarrow \mathsf{KeyGen}_{\mathrm{ot}}(\mathcal{G})$ and set $\Sigma' \leftarrow \mathsf{Sign}(usk, vk'_{\mathrm{ot}})$. [5]
Send the following

1. $Y = G^y, Z = H^y,\ vk'_{\mathrm{ot}},\ \Sigma'$;
2. $\mathbf{c}_M = \mathsf{Com}(ck, M)$; $\mathbf{c}_N := \mathsf{Com}(ck, N)$,
   $\psi_P, \psi_V, \vec{\psi}_\xi$, with $\psi_\odot := \mathsf{CEP}(ck, epk, vk'_{\mathrm{ot}}, \odot)$,
   a proof ϕ_M that $(M, N) \in \mathcal{DH}$ and a proof ϕ_ξ of validity of ξ;
3. $J := (KML^yU^{v'})^{\frac{1}{\eta}}$;
4. a zero-knowledge proof ζ of knowledge of η, y and v' such that
   - $Y = G^y$,
   - $\mathbf{c}_V$ commits to $G^{v'}$, and
   - $\mathbf{c}_M$ commits to $J^\eta L^{-y} U^{-v'} K^{-1}$;
5. $sig' \leftarrow \mathsf{Sign}_{\mathrm{ot}}(sk'_{\mathrm{ot}}, (Y, Z, \Sigma', \mathbf{c}_M, \mathbf{c}_N, \psi_P, \psi_V, \vec{\psi}_\xi, \phi_M, \phi_\xi, J, \zeta, vk'_{\mathrm{ot}}))$.

Issuer If $\Sigma', \psi_P, \psi_V, \vec{\psi}_\xi,\ \phi_M,\ \phi_\xi,\ sig'$ and the proof of knowledge are valid, choose $d, r, v'' \leftarrow \mathbb{Z}_p$ and send:

$$A' := (JT^rU^{v''})^{\frac{1}{x+d}} \quad C := G^d \quad D := F^d \quad R' := G^r \quad S' := H^r \quad v''$$

The user does the following:

1. set $A := (A')^\eta, R := (R')^\eta, S := (S')^\eta, V := G^{v'+\eta v''}, W := H^{v'+\eta v''}$ and check if (A, C, D, R, S) is valid on $\big((M, N), (Y, Z), (V, W)\big)$ under ipk;
2. choose $(vk_{\mathrm{ot}}, sk_{\mathrm{ot}}) \leftarrow \mathsf{KeyGen}_{\mathrm{ot}}$ and define $\Sigma \leftarrow \mathsf{Sign}(y, vk_{\mathrm{ot}})$;

[4] A simulation-sound non-interactive proof of knowledge of Q and W' such that $e(V', F) = e(G, W')$ and $e(P, F) = e(G, Q)$. (cf. §5.2).

[5] The message space for $\mathbf{Sig}$ is the set of DH pairs w.r.t. (G, H'). Since all logarithms of vk_{ot} are known when picking a key, the user can complete the second components of the DH pairs.

3. make commitments $\mathbf{c}_A, \mathbf{c}_C, \mathbf{c}_D, \mathbf{c}_R, \mathbf{c}_S$ to A, C, D, R, S under ck;
4. run $\mathsf{CEP}(ck, epk, vk_{\mathrm{ot}}, \cdot)$ on $Y, Z, \varSigma$; let $\psi_Y, \psi_Z, \vec{\psi}_\varSigma$ denote the outputs;
5. make a proof ϕ_Y that $(Y, Z) \in \mathcal{DH}$ and proofs ϕ_S and $\phi_\varSigma$ of validity of the signatures (A, C, D, R, S) and $\varSigma$;
6. set $sig \leftarrow \mathsf{Sign}_{\mathrm{ot}}\big(sk_{\mathrm{ot}}, (V, W, M, N, \mathbf{c}_A, \mathbf{c}_C, \mathbf{c}_D, \mathbf{c}_R, \mathbf{c}_S, \psi_Y, \psi_Z, \vec{\psi}_\varSigma, \phi_Y, \phi_S, \phi_\varSigma, vk_{\mathrm{ot}})\big)$.

The signature on (M, N) is

$$(V, W, \mathbf{c}_A, \mathbf{c}_C, \mathbf{c}_D, \mathbf{c}_R, \mathbf{c}_S, \psi_Y, \psi_Z, \vec{\psi}_\varSigma, \phi_Y, \phi_S, \phi_\varSigma, vk_{\mathrm{ot}}, sig) \ .$$

Verification. A signature is verified by verifying sig under vk_{ot}, checking the proofs ϕ_Y, ϕ_S and $\phi_\varSigma$, and verifying the encryptions and NIZK proofs in ψ_Y, ψ_Z and $\vec{\psi}_\varSigma$.

Remark 1. As mentioned by [Fuc09], there are two possible instantiations of the zero-knowledge proof of knowledge in Step 4 of User: either using bit-by-bit techniques (which makes the protocol round-optimal); or optimizing the amount of data sent by adding 3 rounds using interactive concurrent Schnorr proofs.

Theorem 2. *The above scheme is an unforgeable blind signature (in the classical sense) under the DLIN, the DHSDH and the HDL assumption.*

The proof of unforgeability is by reduction to unforgeability of Scheme 2, analogously to the proof in [Fuc09]. Note that by additionally extracting y and v' from the proof of knowledge, the simulator can make the special signing queries. The proof of blindness is analogous, too.

Opening of a Transcript ("Signature Tracing"). Given a transcript

$$(Y, Z, \varSigma', \mathbf{c}_M, \mathbf{c}_N, \psi_P, \psi_V, \vec{\psi}_\xi, \phi_M, \phi_\xi, J, \zeta, vk'_{\mathrm{ot}}, sig') \ , \quad v''$$

verify $\varSigma'$, sig', the proofs ϕ_M and ϕ_ξ and the ciphertexts and proofs in ψ_P, ψ_V and $\vec{\psi}_\xi$. If everything is valid, use $rk = ek$ to open the commitments in ψ_P, ψ_V and $\vec{\psi}_\xi$ to P, V' and ξ respectively and set $V := V'P^{v''} = G^{v'+\eta v''}$.

Return $id_\sigma := V$, $upk = Y$ and $\pi := (V', P, v'', \xi, \varSigma')$. The proof π is verified by checking $V = V'P^{v''}$, verifying ξ on V' and P, and verifying $\varSigma'$ under Y.

Opening of a Signature ("Identity Tracing"). Given a *valid* signature

$$(V, W, \mathbf{c}_A, \mathbf{c}_C, \mathbf{c}_D, \mathbf{c}_R, \mathbf{c}_S, \psi_Y, \psi_Z, \vec{\psi}_\varSigma, \phi_Y, \phi_S, \phi_\varSigma, vk_{\mathrm{ot}}, sig) \ ,$$

open the commitments in ψ_Y, ψ_Z and $\vec{\psi}_\varSigma$ using ek and return $upk = Y$ and $\pi = \varSigma$. A proof π is verified by checking if $\varSigma$ is valid on (V, W) under Y.

7 Security Proofs

Theorem 3. *The above scheme is a secure fair blind signature scheme (in the model defined in §2) under the DLIN, the DHSDH and the HDL assumptions.*

Due to space limitation, we sketch the security proofs of all security notions.

Blindness (under DLIN). In the WI setting of GS proofs, commitments and proofs do not reveal anything—and neither do the ciphertexts. Furthermore, for every M and V, there exist η and v' that explain J. In more detail: we proceed by games, Game 0 being the original game. In Game 1, we use the decryption key for the tag-based encryptions to answer tracing queries. Soundness of the NIZK proofs in the ψ's guarantee that the committed and the encrypted values are the same; the games are thus indistinguishable.

In Game 2, we replace the commitment key *ck* by a WI key (indistinguishable under DLIN). In Game 3, we simulate the NIZK proofs in the ψ's and in Game 4, we replace the ciphertexts in the ψ's by encryptions of 0. Games 3 and 4 are indistinguishable by selective-tag weak CCA security of Kiltz' cryptosystem (which follows from DLIN): by unforgeability of the one-time signature, the adversary cannot query a different transcript (or signature) with the same tag as the target transcript (signature), we can thus answer all tracing queries.

In Game 5, we simulate the zero-knowledge proofs in Step 4. In this game, the adversary's view is the following: $J = (KML^y U^{v'})^{\frac{1}{\eta}}$ and M^*, V^* which are either M and $G^{v'+\eta v''}$ or not. Let small letters denote the logarithms of the respective capital letters. Then for every $m^* = \log M^*, v^* = \log V^*$ there exist η, v' such that $v^* = v' + \eta v''$ and $j = \frac{1}{\eta}(k + m^* + yl + v'u)$, i.e. that make M^*, V^* consistent with J. In Game 5, which is indistinguishable from the original game, the adversary has thus no information on whether a given transcript corresponds to a given signature.

Identity Traceability (under DHSDH+HDL). An adversary wins if he can produce a set of valid pairs (m_i, σ_i) s.t. either (I) for one of them the tracing returns $\perp$ or the proof does not verify, or (II) a user appears more often in the openings of the signatures than in the openings of the transcripts. By soundness of Groth-Sahai, we can always extract a user public key and a valid signature. If an adversary wins by (II), then we can use him to forge a $\mathbf{Sig}_3$ signature:

Given parameters and a public key for $\mathbf{Sig}_3$, we set up the rest of the parameters for the blind signature. Whenever the adversary queries his Sign oracle, we do the following: use *ek* to extract (M, N) from $(\mathbf{c}_M, \mathbf{c}_N)$, extract η, y and v' from the zero-knowledge proof of knowledge ζ. Choose $v'' \leftarrow \mathbb{Z}_p$ and query $(M, N, y, v' + \eta v'')$ to signing oracle, receive (A, C, D, R, S) and return $(A^{\frac{1}{\eta}}, C, D, R^{\frac{1}{\eta}}, S^{\frac{1}{\eta}}, v'')$. If the adversary wins by outputting a set of different (i.e. with distinct identifiers (V, W)) blind signatures with one user appearing more often than in the transcripts then among the $\mathbf{Sig}_3$ signatures extracted from the blind signatures there must be a forgery.

Identity Non-Frameability (under DLIN+DHSDH+HDL). Using a successful adversary, we can either forge a signature by the user on vk'_{ot} or a one-time signature (which is secure under DLIN). More precisely, we call an adversary of Type I if it reuses a one-time key from the signatures it received from the User oracle. Since the signature that $\mathcal{A}$ returns must not be contained in *Set*, it is different from the one containing the reused one-time key. The contained one-time signature can thus be returned as a forgery.

An adversary of Type II uses a new one-time key for the returned signature. We use $\mathcal{A}$ to forge a **Sig** signature. The simulator is given parameters (H', K, T) and a public key Y for **Sig**, sets it as one of the honest users' *upk* and queries its signing oracle to simulate the user. Having set $H = G^h$, the simulator can produce $Z = H^y = Y^h$ in the User oracle queries. Since the vk'_{ot} contained $\mathcal{A}$'s output was never queried, we get a valid forgery.

Signature Traceability (under DHSDH+HDL). If the adversary wins by outputting a message/signature pair with an identifier (V, W) s.t. no transcript opens to it, we can extract a $\mathbf{Sig}_3$ signature on (M, N, Y, Z, V, W) without having ever queried a signature on any $(\cdot, \cdot, \cdot, \cdot, V, W)$. The simulation is done analogously to the proof of identity traceability. If the adversary outputs two different signatures they must have different identifiers; one of the ChkSig calls in the experiment returns thus 0. Note that with overwhelming probability two identifiers from different issuing sessions are different (since v'' is chosen randomly by the experiment *after* the adversary chose v' and η).

Signature Non-Frameability (under DLIN+DHSDH+HDL). There are two ways for an issuer to "wrongfully" open a transcript: either he opens it to a user (not necessarily honest) and an identifier of a signature which was produced by an honest user in another session; or it opens to an honest user who has not participated in the issuing session.

Framing an honest signature. Suppose the adversary impersonating the issuer manages to produce a new opening of a transcript that leads to an honestly generated signature. We reduce this framing attack to break CDH, whose hardness is implied by DLIN. Let (G, F, V') be a CDH challenge, i.e. we seek to produce $W' := F^{(\log_G V')}$. Set up the parameters of the scheme setting $H = G^h$ and knowing the trapdoor for SSNIZKPoK. In one of the adversary's User oracle calls, choose $\eta \leftarrow \mathbb{Z}_p$ and use V' from the CDH challenge. Simulate the proof of knowledge of W'. Let v'' be the value returned by the adversary, and let $(V := V'P^\eta, W := V^h)$ be the identifier of the resulting signature.

Suppose the adversary produces a proof $(\bar{V}', \bar{P}, \bar{v}'', \bar{\pi}, \bar{\Sigma})$ with $(\bar{V}', \bar{P}) \neq (V', P)$ for the honest identifier (V, W). By simulation soundness of SSNIZKPoK, we can extract $\bar{W}' = F^{(\log_G \bar{V}')}$ and $\bar{Q} = F^{(\log_G \bar{P})}$. From $V'G^{\eta v''} = V = \bar{V}'\bar{P}^{\bar{v}''}$ we get $V' = \bar{V}'\bar{P}^{\bar{v}''}G^{-\eta v''}$; thus $W' = \bar{W}'\bar{Q}^{\bar{v}''}F^{-\eta v''}$ is a CDH solution. If the adversary recycles (V', P) then it must find a new v'' which leads to a V of an honest signature, and thus has to solve a discrete logarithm.

Framing an honest user. Suppose the adversary outputs an opening of a transcript and a proof revealing an honest user that has never participated in that transcript. Analogously to the proof for signature traceability, we can use the adversary to either forge a signature under a user public key or to forge a one-time signature.

8 Conclusion

We presented the first practical fair blind signature scheme with a security proof in the standard model. The scheme satisfies a new security model strengthening the one proposed by Hufschmitt and Traoré in 2007. The new scheme is efficient (both keys and signatures consist of a constant number of group elements) and does not rely on any new assumptions. As byproducts, we proposed an extension of Fuchsbauer's automorphic signatures, a one-time signature on group elements, and a simulation-sound non-interactive zero-knowledge proof of knowledge of a Diffie-Hellman solution, all three compatible with the Groth-Sahai methodology.

Acknowledgments

This work was supported by the French ANR 07-TCOM-013-04 PACE Project, the European Commission through the IST Program under Contract ICT-2007-216646 ECRYPT II, and EADS.

References

[AF96] Abe, M., Fujisaki, E.: How to date blind signatures. In: Kim, K.-c., Matsumoto, T. (eds.) ASIACRYPT 1996. LNCS, vol. 1163, pp. 244–251. Springer, Heidelberg (1996)

[AO01] Abe, M., Ohkubo, M.: Provably secure fair blind signatures with tight revocation. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 583–602. Springer, Heidelberg (2001)

[BB04] Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)

[BBS04] Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)

[BFI$^+$10] Blazy, O., Fuchsbauer, G., Izabachène, M., Jambert, A., Sibert, H., Vergnaud, D.: Batch Groth-Sahai. Cryptology ePrint Archive, Report 2010/040 (2010), http://eprint.iacr.org/

[BSZ05] Bellare, M., Shi, H., Zhang, C.: Foundations of group signatures: The case of dynamic groups. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 136–153. Springer, Heidelberg (2005)

[CGT06] Canard, S., Gaud, M., Traoré, J.: Defeating malicious servers in a blind signatures based voting system. In: Di Crescenzo, G., Rubin, A. (eds.) FC 2006. LNCS, vol. 4107, pp. 148–153. Springer, Heidelberg (2006)

[Cha83] Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) CRYPTO 1982, pp. 199–203. Plenum Press, New York (1983)

[Dam92] Damgård, I.: Towards practical public key systems secure against chosen ciphertext attacks. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 445–456. Springer, Heidelberg (1992)

[FPV09] Fuchsbauer, G., Pointcheval, D., Vergnaud, D.: Transferable constant-size fair e-cash. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 226–247. Springer, Heidelberg (2009)

[Fuc09] Fuchsbauer, G.: Automorphic signatures in bilinear groups. Cryptology ePrint Archive, Report 2009/320 (2009), http://eprint.iacr.org/

[FV10] Fuchsbauer, G., Vergnaud, D.: Fair blind signatures without random oracles. Cryptology ePrint Archive (2010), Report 2010/101, http://eprint.iacr.org/

[GL07] Groth, J., Lu, S.: A non-interactive shuffle with pairing based verifiability. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 51–67. Springer, Heidelberg (2007)

[Gro06] Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)

[Gro07] Groth, J.: Fully anonymous group signatures without random oracles. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 164–180. Springer, Heidelberg (2007)

[Gro09] Groth, J.: Homomorphic trapdoor commitments to group elements. Cryptology ePrint Archive, Report 2009/007 (2009), http://eprint.iacr.org/

[GS08] Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)

[GT03] Gaud, M., Traoré, J.: On the anonymity of fair offline e-cash systems. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 34–50. Springer, Heidelberg (2003)

[HT07] Hufschmitt, E., Traoré, J.: Fair blind signatures revisited. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 268–292. Springer, Heidelberg (2007)

[Kil06] Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)

[Nao03] Naor, M.: On cryptographic assumptions and challenges (invited talk). In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (2003)

[Oka06] Okamoto, T.: Efficient blind and partially blind signatures without random oracles. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 80–99. Springer, Heidelberg (2006)

[OMA$^+$99] Ohkubo, M., Miura, F., Abe, M., Fujioka, A., Okamoto, T.: An improvement on a practical secret voting scheme. In: Zheng, Y., Mambo, M. (eds.) ISW 1999. LNCS, vol. 1729, pp. 225–234. Springer, Heidelberg (1999)

[RS10] Rückert, M., Schröder, D.: Fair partially blind signatures. In: Bernstein, D.J., Lange, T. (eds.) AFRICACRYPT 2010. LNCS, vol. 6055, pp. 34–51. Springer, Heidelberg (2010)

[SPC95] Stadler, M., Piveteau, J.-M., Camenisch, J.: Fair blind signatures. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 209–219. Springer, Heidelberg (1995)