An extended abstract version of this work appears in: Ron Steinfeld and Philip Hawkes, editors, *Proc. 15th Australasian Conference on Information Security and Privacy (ACISP) 2010*, LNCS. Springer 2010. To appear.
This is the author's version with an enhanced bibliography but is otherwise unchanged from the proceedings version.
EPRINT http://eprint.iacr.org/2010/082.

Predicate-based key exchange

James Birkett Douglas Stebila
Information Security Institute, Queensland University of Technology, Brisbane, Australia
james.birkett@qut.edu.au, douglas@stebila.ca

April 21, 2010

Abstract

We provide the first description of and security model for authenticated key exchange protocols with predicate-based authentication. In addition to the standard goal of session key security, our security model also provides for credential privacy: a participating party learns nothing more about the other party's credentials than whether they satisfy the given predicate. Our model also encompasses attribute-based key exchange since it is a special case of predicate-based key exchange.

We demonstrate how to realize a secure predicate-based key exchange protocol by combining any secure predicate-based signature scheme with the basic Diffie-Hellman key exchange protocol, providing an efficient and simple solution.

Keywords: predicate-based, attribute-based, key exchange, protocols, security models, cryptography

1 Introduction

Two of the fundamental goals of key exchange are authentication and confidentiality. Entity authentication inherently depends on some pre-established piece of trusted information; the most common examples include a shared key, a shared password, or a certified public key. Recently, cryptographers have developed ways of providing more fine-grained access control in cryptographic operations.

Identity-based encryption allows a sender to encrypt a message for a recipient based solely on the recipient's identity (and public parameters for the system); in other words, without requiring a recipient-dependent public key. The identities used in identity-based cryptography may be simple usernames, but they could contain more structured information as well, for example by appending an expiry date or security level. The utility of this idea is limited by the fact that identities must be encoded as strings, and a trusted key generation centre must generate decryption keys for each resulting string.

In attribute-based encryption, a message can be encrypted so that it can only be decrypted by keys whose attributes satisfy a certain policy. Attributes are boolean variables, such as "`student=false`", "`CS_department=false`", and "`Math_department=true`", and policies are boolean functions. Decryption keys are constructed based on the user's attributes, and decryption only succeeds if the user's attributes satisfy the policy encoded in the ciphertext.[1] Research in attribute-based cryptography has focused on encryption and signatures.

The subject of this paper, *predicate-based cryptography*, is a generalization of identity- and attribute-based cryptography. Like attribute-based cryptography, it allows for fine-grained access control based on whether the given credentials satisfy a certain policy. However, credentials and access policies can be more general than in the attribute-based case. Credentials can consist of name-value pairs, where the values can be from arbitrary sets, not just boolean values. Access policies are expressed as predicates

[1] We have described *ciphertext-policy* attribute-based encryption, in which keys have attributes and ciphertexts have policies. These can be switched to obtain *key-policy* attribute-based encryption.

over the set of credentials, and can for example involve equality, comparison, subset, AND, and OR gates. Existing work in predicate-based cryptography has focused on encryption, particularly on expanding the expressiveness of predicates.

Our goal in this work is to consider the use of predicate-based cryptography in a multi-user interactive network setting, specifically examining the cryptographic task of *predicate-based authenticated key exchange*.

1.1 Contributions

Predicate-based key exchange security model. We give the first security model for authenticated key exchange using predicate-based authentication. Our security model has two security experiments:

1. *Session key security*: The session key should be indistinguishable to an adversary. Unlike attribute-based encryption, attribute-based group key exchange, and predicate-based encryption, the session key should be secret even from other parties satisfying the same predicates as either of the two original parties in the key exchange.
2. *Credential privacy*: In a key exchange, it should not be possible for anyone – including the legitimate peer – to learn anything more about a user's credentials other than whether they satisfy the chosen predicate. We argue that this is an essential property for predicate-based key exchange: without it, we might as well return to identity- or public-key-based key exchange with certified lists of credentials.

When restricted to the special case of attribute-based credentials, our security model for predicate-based key exchange also serves as the first full security model for attribute-based key exchange.

A generic predicate-based key exchange protocol. We present a protocol for predicate-based key exchange that satisfies the two security properties above, session key security and attribute privacy. The protocol is a signed-Diffie-Hellman construction that can be used with any secure predicate-based signature scheme. Although our definition of predicate-based signature scheme is new, attribute-based signature schemes are a special case of predicate-based signatures, so attribute-based signatures can be employed in our protocol construction.

Outline. The remainder of this paper is organized as follows. We begin in Sect. 2 with a motivating example. We review existing work in Sect. 3 and introduce notation in Sect. 4. In Sect. 5, we present our security model for predicate-based key exchange protocols – including session key security and attribute privacy – and comment on implementation issues. We define predicate-based signature schemes in Sect. 6, and show in Sect. 7 how to build a secure predicate-based key exchange protocol using predicate-based signatures and a Diffie-Hellman construction. We conclude in Sect. 8.

2 Motivation

When one party wishes to establish a shared secret key with another party, it may not be as simple as Alice saying that she wants to talk to Bob. Alice may in fact wish to talk a customer service supervisor in the international trading group of the Bank of Bob. In other words, Alice has an *policy* against which she checks the *credentials* of the other party. Predicate-based cryptography allows parties to specify fine-grained access control policies and has been used in the context of encryption. It is natural to consider the problem in the context of key exchange, which allow two parties to authentically establish a secure channel.

We begin with a motivating example, drawn from the health care industry. Imagine a patient who wishes to communicate with a psychologist about a mental illness issue. What are some security goals for each party? The goals of the patient are to ensure that she is communicating with a qualified registered psychologist, to use a confidential channel so that no one can eavesdrop, and to maintain her anonymity so her disclosures about her mental illness cannot be used prejudicially against her in another context.

The goals of the psychologist are to verify that the patient has valid insurance coverage from an insurer and to ensure that no one else can eavesdrop on the conversation so as to maintain patient-doctor confidentiality.

There are four types of security goals seen in the example above. The first goal is *policy-based authentication*, where one party can be confident the other party's credentials satisfy some security policy, and moreover that multiple parties cannot *collude* to combine their credentials to satisfy a policy that none of them individually satisfies. The second goal is *confidentiality*, where the parties are ensured that no one except the other authenticated party is able to read their communications; this means only the party with whom we started communicating, not just any partner who satisfies the authentication policy, for we do not want all patients to be able to read messages sent to one patient. The third and fourth goals are interrelated: we seek *anonymity*, so an adversary cannot distinguish between two parties who have credentials satisfying the same policy, and *credential privacy*, meaning that no information is leaked about which precise combination of credentials were used to satisfy the policy.

We aim to achieve these security goals using predicate-based key exchange. The credentials held by a party can be expressed using name-value pairs assigned by one or more credential authorities. For example, a patient with medical insurance may have a private key with the credentials "Employer = Acme Widgets", "Coverage = Gold", "Expires = 2011/06/30", and "Insurer = Red Cross".

The policy used by party to evaluate credentials will be expressed as a *predicate* over credentials; the predicate may be composed of a variety of operations, such as equality and subset tests, AND, OR, and threshold gates, and comparisons. A natural example of a predicate is a *threshold access tree*. Leaves of a threshold access tree consist of boolean-valued functions such as equality tests and comparisons. Interior nodes of a threshold access tree indicate how many of the children nodes must be satisfied; for example, a node with threshold 1 having 4 children corresponds to an OR gate, while a node with threshold n having n children corresponds to an AND gate. An example threshold access tree for the case of a psychologist checking medical insurance is given in Fig. 1.

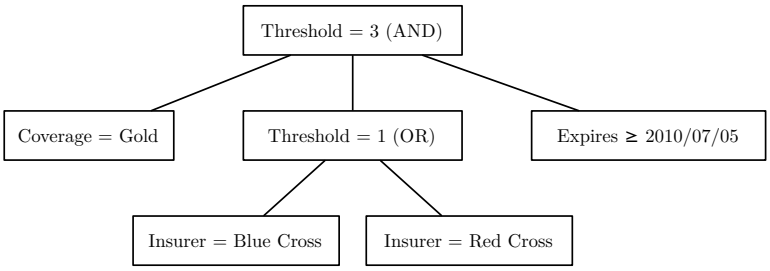

Figure 1: A threshold access tree for checking medical insurance coverage.

3 Related Work

Identity-, attribute-, and predicate-based encryption. *Identity-based encryption*, in which individual parties need not have public keys but only identity strings, was first proposed by Shamir [Sha84] and has recently been the subject of much research. It was extended by Sahai and Waters [SW05] to *fuzzy identity-based encryption* in which parties must match at least a certain number – a threshold – of attributes. An *attribute*, usually labeled by a string, is a boolean variable: it is either present or absent. Goyal et al. [GPSW06] extended fuzzy identity-based encryption to *attribute-based encryption* supporting *boolean threshold access tree* predicates, which consist of boolean combinations of attributes using AND, OR, and threshold gates.

Boneh and Waters [BW07] extended credentials from boolean variable attributes to arbitrary values

and supported encryption using predicates consisting of equality conjunctions, comparison conjunctions, and subset conjunctions; the support of arbitrary, not just boolean, values is what distinguishes *predicate-based* cryptography from attribute-based cryptography. Katz et al. [KSW08] developed a technique for disjunctive predicates and inner products and Shen et al. [SSW09] introduced the notion of predicate privacy for symmetric encryption. The improvement of predicate expressivity continues to be an active area of research.

Key exchange. The first protocol for identity-based key exchange was presented by Günther in 1989 [Gün89] but it was not until 2003 that the first formal security model for identity-based key exchange protocols was proposed by Chen and Kudla [CK03]; their model was an extension of the public key authenticated key exchange security model of Blake-Wilson et al. [BWJM97] (itself based on the Bellare-Rogaway model [BR93]). Kudla and Paterson [KP05] subsequently created a generic key exchange security model to allow for modular security proofs which is also suitable for identity-based key exchange. A more refined security model for identity-based key exchange was proposed by Chen, Cheng, and Smart [CCS07]. A common approach to designing secure key exchange protocols is using a signed-Diffie-Hellman construction (for example, [CK01]).

Wang, Xu, and Ban [WXB09] and Wang, Xu, and Fu [WXF09a, WXF09b] have protocols for what they call attribute-based key agreement protocols (in the random oracle and standard models, respectively). The security proofs treat attributes as identification strings and then revert to the security model of Chen et al. [CCS07] for identity-based authenticated key exchange. These two papers provide no mechanism for evaluating policy predicates and do not consider attribute privacy at all. As such, we consider these schemes to be merely identity-based. Ateniese et al. [AKB07] provide a protocol for secret handshakes – key exchange where participating parties do not learn either the credentials or the predicate of the other party unless the protocol succeeds – using fuzzy attribute matching. Their protocol is secure in the fuzzy selective ID model for encryption [SW05].

Gorantla et al. [GBG10] present a protocol for attribute-based group key exchange, which differs from our work in that all members of the group satisfying the predicate can compute the session key. In contrast, we allow each user to specify a predicate which the peer must satisfy, and these predicates need not be the same; moreover, in our approach the session key can only be computed by the two participants in the key-exchange protocol, not all parties that satisfy the predicate; this is related to the notion of forward-security.

Signature schemes. *Attribute-based signatures* were first introduced by Maji et al. [MPR08], who provided a scheme that supported predicates containing threshold access trees, with a proof in the generic group model. Additional schemes supporting single threshold gates, in either the standard or random oracle models, have been proposed by Shahandashti and Safavi-Naini [SS09] and Li et al. [LAS+10], and a scheme with threshold access trees was given by Khader [Kha08]. These schemes all achieve the goal of *attribute privacy*, in which the attributes used to satisfy a predicate are unknown the verifier. An attribute-based authentication scheme was proposed by Khader et al. [KCD09] with some additional properties beyond signature schemes such as traceability by an authorized entity.

There are also a number of attribute-based group or ring signature schemes that provide lesser privacy guarantees, namely that the signer is anonymous among all signers possessing the same attributes [Kha07b, Kha07a, LK08].

4 Notation

We will use different typefaces to refer to *variables*, algorithms and oracles, and **constants**. The notation $a \leftarrow \mathsf{B}(c)$ indicates that algorithm B is run on input c and the output is assigned to a, and $a \xleftarrow{R} X$ denotes a value x being chosen uniformly at random from the set X. We use the notation $\mathsf{B}(c) \rightarrow a$ and $\mathsf{B}(c) \xrightarrow{R} a$

when defining deterministic and probabilistic algorithms, respectively, with input c and output a. We let $\lambda \in \mathbb{Z}_+$ denote a security parameter. We typically use $\mathcal{A}$ to denote the adversary; $\mathcal{A}^{\mathsf{B}(\cdot)}$ denotes $\mathcal{A}$ run with oracle access to B. Suppose $\mathbb{A}$ is a finite set of size n and $A \in \mathbb{A}$; I_A denotes the binary indicator vector of length n for the set A (assuming a canonical ordering). $\perp$ denotes a null value. We use $\mathbb{G}$ to denote a finite cyclic group, typically of order q and generated by g. A function f is *negligible* if, for sufficiently large x, $|f(x)|$ is smaller than the inverse of any polynomial in x.

Credentials and predicates. Let $\mathbb{C}$ be a finite set; we will call $\mathbb{C}$ the set of *credentials*. A *predicate* is a function $\Phi : \mathbb{C} \to \{\textbf{true}, \textbf{false}\}$. We say that a credential $C \in \mathbb{C}$ *satisfies* a predicate Φ if $\Phi(C) = \textbf{true}$. Let $\mathbb{P} \subseteq \{\textbf{true}, \textbf{false}\}^{\mathbb{C}}$ denote a set of predicates.

5 Predicate-Based Key Exchange

In this section, we define the functionality and security of a predicate-based key exchange protocol.

Definition 1 (Predicate-based key exchange protocol) *Let λ be a security parameter. A* predicate-based key exchange protocol Π *consists of the following algorithms:*

- $\mathsf{Setup}(1^\lambda) \xrightarrow{R} (MPK, MSK)$*: Returns public parameters MPK and a master secret MSK. The public parameters must uniquely define the key space $\mathbb{K}$, the set $\mathbb{C}$ of credentials used in the system and a set $\mathbb{P}$ of predicates over $\mathbb{C}$; we implicitly assume MPK is an input to all subsequent algorithms.*
- $\mathsf{KeyGen}(MSK, C \in \mathbb{C}) \xrightarrow{R} sk$*: The credential issuing authority generates a secret key sk corresponding to the credentials $C \in \mathbb{C}$*
- $\mathsf{Initiate}(sk, role \in \{\textbf{init}, \textbf{resp}\}, \Phi \in \mathbb{P}) \xrightarrow{R} state$*: The user initiates a new session with the given role and predicate Φ.*
- $\mathsf{Action}(sk, m, state) \xrightarrow{R} (m', state, status, k)$*: This is the core of the protocol: it takes a secret key, an incoming message (or the empty string if no messages have yet been exchanged) and the corresponding session state as input and returns the next message in the the protocol, an updated session state, the status of the session (either* **Incomplete**, **Established**, *or* **Failed***), and a session key $k \in \mathbb{K}$, which should be set to $\perp$ until the session is* **Established**.

We have defined predicate-based key exchange in terms of non-interactive algorithms so that it is independent of any networking layer for message delivery. In particular, we deliberately do not specify how the user determines what predicates to use or to which session an incoming message belongs. For example, when using TCP over the Internet, messages may be directed to an IP address (specifying the user) and a port number (specifying the session), but a key-exchange protocol should be substrate-neutral: whether messages are delivered by carrier pigeon or pneumatic tube, the protocol actions are the same. In the case of predicate-based key exchange, these implementation issues have important implications for the security properties we desire, and any application making use of predicate-based key exchange *must* take them into consideration. We will discuss problems that arise from these networking details further in Sect. 5.3.

5.1 Correctness

A predicate-based key exchange is correct if, whenever two users who each satisfy their peer's predicate run the protocol over a benign network which faithfully delivers their messages unaltered, both parties complete the session in state **Established** and they agree on a key.

Let $role(j) = R$ if j is even and $role(j) = I$ if j is odd. Let $\mathsf{Correct}(MSK, C_I, C_R, \Phi_I, \Phi_R)$ be as follows: Set $sk_I \leftarrow \mathsf{KeyGen}(MSK, C_I)$ and $sk_R \leftarrow \mathsf{KeyGen}(MSK, C_R)$. Let $state_I \leftarrow \mathsf{Initiate}(sk_I, \textbf{init}, \Phi_I)$, and $state_R \leftarrow \mathsf{Initiate}(sk_R, \textbf{resp}, \Phi_R)$. Set $(m_1, state_I, status_I, k) \leftarrow \mathsf{Action}(sk_I, \perp, state_I)$. For $j = 1, \ldots, r-1$, set $(m_{j+1}, state_{role(j+1)}, status_{role(j+1)}, k_{role(j+1)}) \leftarrow \mathsf{Action}(sk_{role(j)}, m_j, state_{role(j)})$. If $status_I = \textbf{Established} = status_R$ and $k_I = k_R$, then return **true**, otherwise return **false**.

Definition 2 (Correctness) *A predicate-based key-exchange protocol is said to be* correct *if, for* $(MPK, MSK) \leftarrow \mathsf{Setup}(1^k)$*, for all* $\Phi_I, \Phi_R \in \mathbb{P}$ *and for all* $C_I, C_R \in \mathbb{C}$ *such that* $\Phi_R(C_I) = \mathbf{true} = \Phi_I(C_R)$*,*

$$\Pr(\mathsf{Correct}(MSK, C_I, C_R, \Phi_I, \Phi_R) = \mathbf{true}) = 1 \ .$$

5.2 Security Model

We require a predicate-based key exchange protocol to satisfy two security properties: session-key security and credential privacy. Our security model combines aspects of the Bellare-Rogaway [BR93] model for key exchange, the Maji et al. model for attribute-based signature schemes [MPR08], and aspects of predicate-based encryption from Boneh and Waters [BW07]. We define these properties using two security experiments, each played by an adversary against a challenger.

In both security experiments, the challenger maintains a list of users $U_1, \ldots, U_N$, which is not fixed, but is under the control of the adversary. Each user U_u has credentials C_u and a secret key sk_u, and the challenger maintains a numbered list of sessions, $s_{u,1}, \ldots, s_{u,n_u}$, with the following associated variables:

- $m_{u,\ell,1}, \ldots, m_{u,\ell,i}$: The protocol messages exchanged in session $s_{u,\ell}$.
- $state_{u,\ell}$: The private session state information.
- $status_{u,\ell} \in \{\mathbf{Established}, \mathbf{Incomplete}, \mathbf{Failed}\}$: The status of the session.
- $k_{u,\ell} \in \mathbb{K}$: The session key.
- $\Phi_{u,\ell} \in \mathbb{P}$: The predicate which the peer of the session must satisfy.
- $\Phi'_{u,\ell} \in \mathbb{P}$: The predicate which the owner of the session must satisfy; in our example construction, this value is sent to the peer as part of the first protocol message, but it could in principle be specified by some other means.
- $role_{u,\ell} \in \{\mathbf{init}, \mathbf{resp}\}$: The role (initiator or responder) played by the user U_u in session ℓ.

We now present the queries available to the adversary in both games:

- $\mathsf{Create}(C \in \mathbb{C})$: The challenger increments N, the number of users, sets $C_N \leftarrow C$, computes $sk_N \leftarrow \mathsf{KeyGen}(MSK, C_N)$ and returns N.
- $\mathsf{Activate}(u, role, \Phi \in \mathbb{P})$: The challenger increments n_u, sets $state_{u,n_u} \leftarrow \mathsf{Initiate}(sk_u, role, \Phi)$, and returns n_u.
- $\mathsf{Send}(u, \ell, m_{u,\ell,i})$: The challenger sets $(m_{u,\ell,i+1}, state_{u,\ell}, status_{u,\ell}, k_{u,\ell}) \leftarrow \mathsf{Action}(sk_u, m_{u,\ell}, state_{u,\ell})$ and returns $(m_{u,\ell,i+1}, status_{u,\ell})$. If $role_{u,\ell} = \mathbf{init}$ and $i = 0$, then $m_{u,\ell,i}$ must be $\perp$.
- $\mathsf{SKReveal}(u, l)$: Returns $k_{u,\ell}$.
- $\mathsf{Corrupt}(u)$: Returns sk_u.

5.2.1 Session Key Security.

The definition of session key security is based on the idea that an adversary should not be able to distinguish the session key of a sufficiently uncompromised session from a random string, except with negligible probability. First, we adapt the Bellare-Rogaway definition of a matching conversation [BR93] to our setting as follows.

Definition 3 (Matching session) *A session* $s_{u',\ell'}$ is a matching session of *a session* $s_{u,\ell}$ *if* $\Phi_{u,\ell} = \Phi'_{u',\ell'}$, $\Phi'_{u,\ell} = \Phi_{u',\ell'}$*, and any of the following rules hold.*

- *For protocols where* r*, the number of rounds, is odd:*
  - $role_{u,\ell} = \mathbf{init}$*,* $role_{u',\ell'} = \mathbf{resp}$*, and* $(m_{u,\ell,1}, \ldots, m_{u,\ell,r-1}) = (m_{u',\ell',1}, \ldots, m_{u',\ell',r-1})$*;*
  - $role_{u,\ell} = \mathbf{resp}$*,* $role_{u',\ell'} = \mathbf{init}$*, and* $(m_{u,\ell,1}, \ldots, m_{u,\ell,r}) = (m_{u',\ell',1}, \ldots, m_{u',\ell',r})$*.*
- *For protocols where* r *is even:*

- $role_{u,\ell} = \textbf{init}$, $role_{u',\ell'} = \textbf{resp}$, *and* $(m_{u,\ell,1}, \ldots, m_{u,\ell,r}) = (m_{u',\ell',1}, \ldots, m_{u',\ell',r})$;
- $role_{u,\ell} = \textbf{resp}$, $role_{u',\ell'} = \textbf{init}$, *and* $(m_{u,\ell,1}, \ldots, m_{u,\ell,r-1}) = (m_{u',\ell',1}, \ldots, m_{u',\ell',r-1})$.

This captures the idea that the owner and the peer in the matching session must satisfy each other's predicates and agree on all of the messages exchanged, except perhaps if the owner of the session $s_{u,\ell}$ sent the final message. In this case the owner of the session completes the protocol without knowing if the final message was delivered, or if a different message was delivered instead, so we do not require that the final messages are equal in this case. Note that the relation "is a matching session of" is *not* symmetric!

Definition 4 (Session key security) *Let λ be a security parameter and let $\mathcal{A}$ be a polynomial-time (in λ) probabilistic algorithm. A predicate-based key exchange protocol Π is* session-key-secure *if*

$$\mathsf{Adv}^{\text{PB-SK}}_{\Pi,\mathcal{A}}(\lambda) := \left| \Pr\left(\mathsf{Expt}^{\text{PB-SK}}_{\Pi,\mathcal{A}}(\lambda) = \textbf{true} \right) - \frac{1}{2} \right|$$

is negligible, where $\mathsf{Expt}^{\text{PB-SK}}_{\Pi,\mathcal{A}}(\lambda)$ is the following algorithm:

1. *Set $(MPK, MSK) \leftarrow \mathsf{Setup}(1^\lambda)$.*
2. *Let $\mathsf{Test}(u,\ell)$ be the following algorithm. Choose a bit $b \xleftarrow{R} \{0,1\}$ at random. If $b = 0$, then return $k_{u,\ell}$, otherwise return $k \xleftarrow{R} \mathbb{K}$.*
3. *Set $b' \leftarrow \mathcal{A}(MPK)$, where $\mathcal{A}$ has oracle access to* Create, Activate, Send, SKReveal, Corrupt, *and* Test. *$\mathcal{A}$ is restricted as follows:*
   - *$\mathcal{A}$ may make a single query to the* Test *oracle; let u, ℓ be the arguments to that query.*
   - *$\mathcal{A}$ must not have made any query of the form $\mathsf{Corrupt}(u')$ for any u' such that $\Phi_{u,\ell}(C_{u'}) = \textbf{true}$ prior to the* Test *query.*
   - *When the* Test *query is made, it must be that $status_{u,\ell} = \textbf{Established}$.*
   - *$\mathcal{A}$ may not query $\mathsf{SKReveal}(u,\ell)$ or $\mathsf{SKReveal}(u',\ell')$ for any (u',ℓ') such that $s_{u',\ell'}$ is a matching session of $s_{u,\ell}$, even after the* Test *query is made.*
4. *If $b' = b$, then return* **true**, *otherwise return* **false**.

Collusion resistance. This definition of session key security also implies collusion resistance, since the adversary may perform Corrupt queries for multiple users with credentials that collectively, but not individually, satisfy the predicate.

5.2.2 Credential Privacy.

For the credential privacy experiment, the adversary should not be able to distinguish between two users whose credentials satisfy the same predicate, even if they have different credentials.

Definition 5 (Credential privacy) *Let λ be a security parameter and let $\mathcal{A}$ be a polynomial-time (in λ) probabilistic algorithm. A predicate-based key exchange protocol Π is* credential-private *if*

$$\mathsf{Adv}^{\text{PB-Priv}}_{\Pi,\mathcal{A}}(\lambda) := \left| \Pr\left(\mathsf{Expt}^{\text{PB-Priv}}_{\Pi,\mathcal{A}}(\lambda) = \textbf{true} \right) - \frac{1}{2} \right|$$

is negligible, where $\mathsf{Expt}^{\text{PB-Priv}}_{\Pi,\mathcal{A}}(\lambda)$ is the following algorithm:

1. *Set $(MPK, MSK) \leftarrow \mathsf{Setup}(1^\lambda)$.*

2. *Let* $\mathsf{TestActivate}(u_0, u_1, role, \Phi \in \mathbb{P})$ *be the following algorithm. Choose a bit* $b \xleftarrow{R} \{0,1\}$ *at random. Set* $state^* \leftarrow \mathsf{Initiate}(sk_{u_b}, role, \Phi)$ *and return* $\perp$.
3. *Let* $\mathsf{Send}^*(m_i^*)$ *be the following algorithm. Set* $(m_{i+1}^*, state^*, status^*, k^*) \leftarrow \mathsf{Action}(sk_{u_b}, m_i^*, state^*)$ *and return* m_{i+1}^*.
4. *Set* $b' \leftarrow \mathcal{A}(MPK)$*, where* $\mathcal{A}$ *has oracle access to* Create*,* Activate*,* Send*,* Send^**,* SKReveal*,* Corrupt*, and* TestActivate*.* $\mathcal{A}$ *is restricted as follows:*
   - $\mathcal{A}$ *may make a single query to the* TestActivate *oracle.*
   - *The predicate* Φ'^* *which* C_{u_b} *has to satisfy (which is determined by the* $\mathsf{Send}^*(\cdot)$ *queries made by the adversary) must be chosen so that* $\Phi'^*(C_{u_0}) = \Phi'^*(C_{u_1})$*. (If this were not the case then the adversary could trivially distinguish* U_{u_0} *from* U_{u_1}*.)*
5. *If* $b' = b$*, then return* **true***, otherwise return* **false***.*

Credential privacy captures the notion of anonymity: the adversary cannot distinguish between two users satisfying the same predicate. It also ensures that the adversary cannot tell whether two sessions with the same predicate are owned by the same user; we call this property *unlinkability*. To see why this holds, suppose that an adversary executes a session with U_{u_0}, and the test session with U_{u_b} using the same predicate. If the adversary *could* tell whether those two sessions are owned by the same user, then it can discover the identity of U_{u_b} and win the credential privacy experiment.

5.3 Implementation Issues

Credential privacy is an essential feature of any predicate-based key exchange protocol. If an application does not need credential privacy, then standard public key or identity-based systems may be used in combination with a credential-issuing authority which simply issues a certificate on the users public key declaring that they hold a given credential. This shows that there is simply no need for predicate-based key exchange unless credential-privacy is desired.

Our definition of credential privacy ensures that the contents of the protocol messages exchanged reveal no information about either party's credentials, except whether they satisfy their peer's chosen predicate. Unlike predicate-based encryption or signatures, predicate-based key exchange faces an additional challenge: users need to be identified by some means in order to deliver messages. It seems unavoidable that this should leak some information about a user's credentials, but we will discuss some approaches that may be fruitful.

Suppose that a predicate-based key exchange protocol is used on an IP network, with each user having a fixed IP address. An adversary may initiate multiple sessions with the same user using different predicates to exhaustively search the credential space. A user initiating a session may mitigate this problem if she is able to obtain a new IP address for each session, for example by using tunnelling, or an anonymising service such as Tor [DMS04]. Unfortunately, a user acting as a responder cannot use this solution, since the initiator must know an address to initiate a session. Depending on the application, it may be that only the initiator needs credential privacy. In the example from Sect. 2, the patient desires to remain anonymous when discussing their mental-health problems, but it seems unlikely that the psychologist has the same requirement. However, a society of secretive psychologists acting together could preserve some degree of anonymity by operating a trusted proxy which knows their individual credentials, and could choose a psychologist who satisfies a given predicate at random from among the society.

6 Predicate-Based Signature Schemes

Our definition of predicate-based signature schemes is a natural extension from the definition of attribute-based signature schemes [MPR08].

Definition 6 (Predicate-based signature scheme) *Let λ be a security parameter. A* predicate-based signature scheme $\mathcal{S}$ *is a tuple consisting of the following polynomial-time (in λ) algorithms:*

- $\mathsf{Setup}(1^\lambda) \xrightarrow{R} (mpk, msk)$*: The credential authority obtains a master private key msk and public parameters mpk. The public parameters must uniquely define the set $\mathbb{C}$ of credentials and a set $\mathbb{P}$ of predicates over $\mathbb{C}$; we assume mpk is an implicit input to all subsequent algorithms.*
- $\mathsf{KeyGen}(msk, C \in \mathbb{C}) \xrightarrow{R} sk$*: The authority generates a signing key sk for credentials C.*
- $\mathsf{Sign}(sk, m, \Phi \in \mathbb{P}) \xrightarrow{R} \sigma$*: The signer generates a signature σ for a message m and predicate Φ, provided sk was generated with C such that $\Phi(C) =$* **true**.
- $\mathsf{Verify}(m, \Phi \in \mathbb{P}, \sigma) \to \{$**true**, **false**$\}$*: The verifier checks if σ is a valid signature on m for predicate Φ.*

Definition 7 (Correctness) *A predicate-based signature scheme $\mathcal{S}$ is* correct *if, for $(mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda)$, all messages m, all credentials $C \in \mathbb{C}$, all signing keys $sk \leftarrow \mathsf{KeyGen}(msk, C)$, and all predicates $\Phi \in \mathbb{P}$ such that $\Phi(C) =$* **true***, we have* $\Pr(\mathsf{Verify}(m, \Phi, \mathsf{Sign}(sk, m, \Phi)) =$ **true**$) = 1$.

Definition 8 (Perfect privacy) *A predicate-based signature scheme $\mathcal{S}$ is* perfectly private *if, for $(mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda)$, all messages m, all credentials $C_1, C_2 \in \mathbb{C}$, all signing keys $sk_1 \leftarrow \mathsf{KeyGen}(msk, C_1)$, $sk_2 \leftarrow \mathsf{KeyGen}(msk, C_2)$, and all predicates $\Phi \in \mathbb{P}$ such that $\Phi(C_1) = \Phi(C_2) =$* **true***, the distributions $\mathsf{Sign}(sk_1, m, \Phi)$ and $\mathsf{Sign}(sk_2, m, \Phi)$ are equal.*

A perfectly private predicate-based signature scheme does not leak any information about which credentials or secret keys were used in signing.

Definition 9 (Unforgeability) *Let λ be a security parameter and let $\mathcal{A}$ be a polynomial-time (in λ) probabilistic algorithm. A perfectly private predicate-based signature scheme $\mathcal{S}$ is* unforgeable *if*

$$\mathsf{Adv}^{\text{PB-Forge}}_{\mathcal{S},\mathcal{A}}(\lambda) := \Pr\left(\mathsf{Expt}^{\text{PB-Forge}}_{\mathcal{S},\mathcal{A}}(\lambda) = \textbf{true}\right)$$

is negligible, where $\mathsf{Expt}^{\text{PB-Forge}}_{\mathcal{S},\mathcal{A}}(\lambda)$ is the following algorithm:

1. *Set $(mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda)$.*
2. *Let $\mathsf{AltSign}(msk, m, C \in \mathbb{C}, \Phi \in \mathbb{P})$ be an algorithm that, provided $\Phi(C) =$* **true***, sets $sk \leftarrow \mathsf{KeyGen}(msk, C)$, and returns $\mathsf{Sign}(sk, m, \Phi)$.*
3. *Set $(m, \Phi, \sigma) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(msk,\cdot),\mathsf{AltSign}(msk,\cdot,\cdot)}(mpk)$.*
4. *If $\mathsf{Verify}(m, \Phi, \sigma) =$* **true***, $\mathcal{B}$ never queried $\mathsf{AltSign}(m, \cdot, \Phi)$, and $\mathcal{B}$ never queried $\mathsf{KeyGen}(C)$ for any $C \in \mathbb{C}$ such that $\Phi(C) =$* **true***, then return* **true***, otherwise return* **false**.

The security experiment for unforgeability is slightly different than is typical for signature schemes, because the signing oracle generates a new key for each signature rather than using an existing key. However, for a predicate-based signature scheme with perfect privacy, the signature depends on the predicate used, but not the specific credentials (or secret key), so the definition is appropriate.

An example instantiation. Attribute-based signature schemes are a special case of predicate-based signature schemes. We can rewrite the notation of attribute-based signature schemes in terms of the more expressive notation of predicate-based schemes, as indicated in Fig. 2. Thus, all attribute-based schemes are predicate-based schemes, but in general predicate-based schemes are more expressive than attribute-based schemes. It follows that existing secure attribute-based schemes [MPR08, SS09, KCD09] are also secure predicate-based signature schemes.

	Attribute-based [MPR08]	Predicate-based (Sect. 4)
Credential universe	$\mathbb{A}$, $\lvert\mathbb{A}\rvert = n$	$\mathbb{C} = \{0,1\}^{\lvert\mathbb{A}\rvert}$
Credentials	$A \subseteq \mathbb{A}$	$C \in \mathbb{C}$, $C = I_A$
Predicate	$\Upsilon : \{0,1\}^n \to \{\textbf{true}, \textbf{false}\}$ A satisfies Υ iff $\Upsilon(I_A) = \textbf{true}$	$\Phi : \mathbb{C} \to \{\textbf{true}, \textbf{false}\}$ C satisfies Φ iff $\Phi(C) = \textbf{true}$

Figure 2: Representation of attribute-based notation in predicate-based notation.

7 A Signed Diffie-Hellman Construction

We present a simple signed-Diffie-Hellman protocol using a secure predicate-based signature scheme and a group in which the Decisional Diffie-Hellman (DDH) problem is hard.

Definition 10 (Decisional Diffie-Hellman problem [Bon98]) *Let $(\mathbb{G}_\lambda)_{\lambda\in\mathbb{N}}$ be a family of multiplicatively written cyclic groups of prime order q_λ, indexed by a security parameter λ. Fix a security parameter λ; let g be a generator of $\mathbb{G}_\lambda$ and let $x, y, z \xleftarrow{R} \mathbb{Z}_{q_\lambda}$. For any probabilistic polynomial-time algorithm $\mathcal{A}$, we define*

$$\mathsf{Adv}^{\mathrm{DDH}}_{\mathbb{G}_\lambda,\mathcal{A}}(\lambda) = \left|\Pr\left(\mathcal{A}(g, g^x, g^y, g^z) = 1\right) - \Pr\left(\mathcal{A}(g, g^x, g^y, g^{xy}) = 1\right)\right| \ .$$

The DDH problem is hard *if, for any probabilistic polynomial-time algorithm $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{DDH}}_{\mathbb{G}_\lambda,\mathcal{A}}(\lambda)$ is negligible.*

7.1 Protocol Definition

Let $\mathcal{S} = (\mathsf{Setup}_\mathcal{S}, \mathsf{KeyGen}_\mathcal{S}, \mathsf{Sign}, \mathsf{Verify})$ be a predicate-based signature scheme. We define the protocol $\Pi_{\mathcal{S},\mathbb{G}}$ as the following tuple of algorithms:

- $\mathsf{Setup}(1^\lambda)$: Set $(mpk, msk) \leftarrow \mathsf{Setup}_\mathcal{S}(1^\lambda)$; recall that mpk defines a set $\mathbb{C}$ of credentials and a set $\mathbb{P}$ of predicates over $\mathbb{C}$. Let $\mathbb{G} = \mathbb{G}_\lambda$ be a finite cyclic group of order $q = q_\lambda$ generated by g. Set $MPK \leftarrow (mpk, \mathbb{G}, g, q)$ and $MSK \leftarrow msk$. Return (MPK, MSK).
- $\mathsf{KeyGen}(MSK, C \in \mathbb{C})$: Return $\mathsf{KeyGen}_\mathcal{S}(msk, C)$.
- $\mathsf{Initiate}(sk, \textbf{init}, \Phi_I)$: Return $state \leftarrow \Phi_I$.
- $\mathsf{Initiate}(sk, \textbf{resp}, \Phi_R)$: Return $state \leftarrow \Phi_R$.
- $\mathsf{Action}(sk, m, state)$: For clarity, we write the protocol action as four separate algorithms which may be combined in the natural way. We also present the protocol diagrammatically in Fig. 3.
  - $\mathsf{InitiatorAction1}(sk, \bot, \Phi_I)$: Set $x \xleftarrow{R} \mathbb{Z}_q$ and $X \leftarrow g^x$. Set $m' \leftarrow (X, \Phi_I)$ and $state' \leftarrow (\Phi_I, x)$. Return $(m', state', \textbf{Incomplete}, \bot)$.
  - $\mathsf{ResponderAction1}(sk, (X, \Phi_I), \Phi_R)$: If $\Phi_I(C_R) = \textbf{false}$, then return $(\bot, \bot, \textbf{Failed}, \bot)$. Otherwise, set $y \xleftarrow{R} \mathbb{Z}_q$ and $Y \leftarrow g^y$. Set $\sigma_R \leftarrow \mathsf{Sign}(sk, (\textbf{resp}, X, \Phi_I, Y, \Phi_R), \Phi_I)$. Set $m' \leftarrow (Y, \Phi_R, \sigma_R)$ and $state' \leftarrow (X, \Phi_I, Y, y, \Phi_R, \sigma_R)$. Return $(m', state', \textbf{Incomplete}, \bot)$.
  - $\mathsf{InitiatorAction2}(sk, (Y, \Phi_R, \sigma_R), (\Phi_I, x))$: If $\mathsf{Verify}((\textbf{resp}, X, \Phi_I, Y, \Phi_R), \Phi_I, \sigma_R) = \textbf{false}$ or $\Phi_R(C_I) = \textbf{false}$, then return $(\bot, \bot, \textbf{Failed}, \bot)$. Set $\sigma_I \leftarrow \mathsf{Sign}(sk, (\textbf{init}, X, \Phi_I, Y, \Phi_R, \sigma_R), \Phi_R)$. Set $k \leftarrow Y^x$. Return $(\sigma_I, \bot, \textbf{Established}, k)$.
  - $\mathsf{ResponderAction2}(sk, \sigma_I, (X, \Phi_I, Y, y, \Phi_R, \sigma_R))$: If $\mathsf{Verify}((\textbf{init}, X, \Phi_I, Y, \Phi_R, \sigma_R), \Phi_R, \sigma_I) \neq \textbf{true}$, then return $(\bot, \bot, \textbf{Failed}, \bot)$. Set $k \leftarrow X^y$. Return $(\bot, \bot, \textbf{Established}, k)$.

It is easy to see that the $\Pi_{\mathcal{S},\mathbb{G}}$ is correct when the signature scheme is correct.

7.2 Credential Privacy

Theorem 1 *If $\mathcal{S}$ is a perfectly-credential-private signature scheme, then $\Pi_{\mathcal{S},\mathbb{G}}$ is credential-private.*

$\Pi_{\mathcal{S},\mathbb{G}}$ – Protocol flow		
Initiator secret key sk_I responder predicate Φ_I		**Responder** secret key sk_R initiator predicate Φ_R
InitiatorAction1 $x \xleftarrow{R} \mathbb{Z}_q, X \leftarrow g^x$	$\xrightarrow{X,\Phi_I}$	ResponderAction1 $y \xleftarrow{R} \mathbb{Z}_q, Y \leftarrow g^y$
InitiatorAction2 If $\neg\mathsf{Verify}((\mathbf{resp},X,\Phi_I,Y,\Phi_R),\Phi_I,\sigma_R)$ then $status \leftarrow$ **Failed** Abort $\sigma_I \leftarrow \mathsf{Sign}(sk_I,(\mathbf{init},X,\Phi_I,Y,\Phi_R,\sigma_R),\Phi_R)$ $k \leftarrow Y^x$ $status \leftarrow$ **Established**	$\xleftarrow{Y,\Phi_R,\sigma_R}$ $\xrightarrow{\sigma_I}$	$\sigma_R \leftarrow \mathsf{Sign}(sk_R,(\mathbf{resp},X,\Phi_I,Y,\Phi_R),\Phi_I)$ ResponderAction2 If $\neg\mathsf{Verify}((\mathbf{init},X,\Phi_I,Y,\Phi_R,\sigma_R),\Phi_R,\sigma_I)$ then $status \leftarrow$ **Failed** Abort $k \leftarrow X^y$ $status \leftarrow$ **Established**

Figure 3: Protocol flow of $\Pi_{\mathcal{S},\mathbb{G}}$.

PROOF. [sketch] Consider the test session in the credential privacy experiment for the predicate-based key exchange protocol. If u_b does not satisfy the chosen predicate Φ'^*, specified by the adversary – that is, if $\Phi'^*(C_{u_b}) =$ **false** – then the session terminates with status **Failed**, by definition of the protocol. However, the choice of Φ'^* is restricted so that $\Phi'^*(C_{u_0}) = \Phi'^*(C_{u_1})$, so in this case the responses of the challenger are independent of the bit b. Similarly, if $\Phi'^*(C_{u_b}) =$ **true**, the distribution of the signature returned to the adversary does not depend on the bit b by the perfect privacy of $\mathcal{S}$. Since the bit b is not used in answering any other queries, we now see that the responses to the adversary's queries are all independent of b, so $\Pr(b' = b) = \frac{1}{2}$ and $\mathsf{Adv}^{\text{PB-Priv}}_{\Pi_{\mathcal{S},\mathbb{G}},\mathcal{A}}(\lambda) = 0$. □

7.3 Session Key Security

Theorem 2 *If $\mathcal{S}$ is an unforgeable signature scheme and the DDH problem is hard in $\mathbb{G}$, then $\Pi_{\mathcal{S},\mathbb{G}}$ is session-key secure.*

PROOF. Let $\mathcal{A}$ be an adversary against the session key security of $\Pi_{\mathcal{S},\mathbb{G}}$ and consider the experiment $\mathsf{Expt}^{\text{PB-SK}}_{\Pi_{\mathcal{S},\mathbb{G}}}(\lambda)$. Let u^*, ℓ^* be the test session. Define M to be the event that a matching session $s_{u',\ell'}$ of s_{u^*,ℓ^*} exists.

Case 1: No session matching s_{u^*,ℓ^*} exists (event $\neg M$). We construct an adversary $\mathcal{B}$ against the unforgability of $\mathcal{S}$ as follows. $\mathcal{B}$ runs $\mathcal{A}(mpk)$ and simulates the challenger's responses according to the definition of the $\mathsf{Expt}^{\text{PB-SK}}_{\Pi_{\mathcal{S},\mathbb{G}},\lambda}$, with the following modifications: whenever the challenger would compute $\mathsf{Sign}(sk_u, m, \Phi)$ (while responding to a Send query), $\mathcal{B}$ instead queries the AltSign oracle on input (msk, m, C_u, Φ). Whenever $\mathcal{A}$ makes a $\mathsf{Corrupt}(u)$ query, $\mathcal{B}$ responds by querying $\mathsf{KeyGen}_{\mathcal{S}}(C_u)$ and returning the result.

Now consider the test session s_{u^*,ℓ^*}. By the definition of $\Pi_{\mathcal{S},\mathbb{G}}$, $m_{u^*,\ell^*,1} = (X, \Phi_{u^*,\ell^*})$ for some $X \in \mathbb{G}$, $m_{u^*,\ell^*,2} = (Y, \Phi'_{u^*,\ell^*}, \sigma_R)$ for some $Y \in \mathbb{G}$, and $m_{u^*,\ell^*,3} = \sigma_I$. When $\mathcal{A}$ terminates, if $role_{u^*,\ell^*} =$ **init**, then $\mathcal{B}$ chooses $m^* \leftarrow (\mathbf{resp}, X, \Phi_{u^*,\ell^*}, Y, \Phi'_{u^*,\ell^*})$ as the message to forge a signature on and returns $(m^*, \Phi_{u^*,\ell^*}, \sigma_R)$ as the forgery. If $role_{u^*,\ell^*} =$ **resp**, $\mathcal{B}$ chooses $m^* \leftarrow (\mathbf{init}, X, \Phi'_{u^*,\ell^*}, Y, \Phi_{u^*,\ell^*}, \sigma_R)$ and

returns $(m^*, \Phi_{u^*,\ell^*}, \sigma_I)$ as the forgery.

We must now show that if the test session has no matching session, then $\mathcal{B}$ satisfies the requirements of Definition 9, namely that $\mathsf{Verify}(m, \Phi, \sigma) = \textbf{true}$, $\mathcal{B}$ never queried $\mathsf{AltSign}(msk, m, \cdot, \Phi)$ and $\mathcal{B}$ never queried $\mathsf{KeyGen}_{\mathcal{S}}(C)$ for any credential C such that $\Phi(C) = \textbf{true}$.

Since the test session must be an **Established** session, it follows that $\mathsf{Verify}(m, \Phi_{u^*,\ell^*}, \sigma_R) = \textbf{true}$. Because of the constraints on $\mathcal{A}$ concerning the test session, it follows that $\mathcal{A}$ never queried $\mathsf{Corrupt}(u)$ for any u satisfying $\Phi_{u^*,\ell^*}(C_u) = \textbf{true}$, which implies that $\mathcal{B}$ never queried $\mathsf{KeyGen}_{\mathcal{S}}(C)$ for any credential C such that $\Phi_{u^*,\ell^*}(C) = \textbf{true}$.

Finally, suppose $\mathcal{A}$ made a query of the form $\mathsf{Send}(u', \ell', m_{u',\ell',i})$ which caused $\mathcal{B}$ to query $\mathsf{AltSign}(m^*, C, \Phi_{u^*,\ell^*})$, where m^* is the forged message defined above.

If $role_{u^*,\ell^*} = \textbf{init}$, then $m^* = (\textbf{resp}, X, \Phi_{u^*,\ell^*}, Y, \Phi'_{u^*,\ell^*})$, and the only circumstances where $\mathcal{B}$ could query $\mathsf{AltSign}(msk, m^*, C, \Phi_{u^*,\ell^*})$ are if $\Phi_{u',\ell'} = \Phi'_{u^*,\ell^*}$, $\Phi'_{u',\ell'} = \Phi_{u^*,\ell^*}$, $m_{u',\ell',1} = (X, \Phi'_{u',\ell'})$, and $m_{u',\ell',2} = (Y, \Phi_{u',\ell'}, \sigma_R)$: in other words, when $s_{u',\ell'}$ is a matching session of s_{u^*,ℓ^*}, contradicting our original assumption. Conversely, if $role_{u^*,\ell^*} = \textbf{resp}$, then $m^* = (\textbf{init}, X, \Phi'_{u^*,\ell^*}, Y, \Phi_{u^*,\ell^*}, \sigma_R)$, and if $\mathcal{B}$ queried $\mathsf{AltSign}(m^*, C, \Phi_{u^*,\ell^*})$ then $\Phi_{u',\ell'} = \Phi'_{u^*,\ell^*}$, $\Phi'_{u',\ell'} = \Phi_{u^*,\ell^*}$, $m_{u',\ell',1} = (X, \Phi_{u',\ell'})$, $m_{u',\ell',2} = (Y, \Phi'_{u',\ell'}, \sigma_R)$ and $m_{u',\ell',3} = \sigma_I$. Once again this implies that $s_{u',\ell'}$ is a matching session of s_{u^*,ℓ^*} contradicting our original assumption.

Therefore $\mathcal{B}$ wins the forgery game whenever $\mathcal{A}$ selects a test session with no matching session, so $\Pr(\neg M) = \mathsf{Adv}^{\text{PB-Forge}}_{\mathcal{S},\mathcal{B}}(\lambda)$, which is negligible.

Case 2: There is a session $s_{u',\ell'}$ which matches s_{u^*,ℓ^*} (event M). Since s_{u^*,ℓ^*} is required to be **Established**, and $s_{u',\ell'}$ matches s_{u^*,ℓ^*} by assumption, we see that $m_{u^*,\ell^*,1} = (X, \Phi_{u^*,\ell^*}) = (X, \Phi'_{u',\ell'}) = m_{u',\ell',1}$, $m_{u^*,\ell^*,2} = (Y, \Phi_{u',\ell'}, \sigma_R) = (Y, \Phi'_{u^*,\ell^*}, \sigma_R) = m_{u',\ell',2}$.

In particular, this shows that both X and Y were chosen by the challenger in response to the corresponding Send queries. This allows us to construct a DDH adversary $\mathcal{C}$ as follows. Let $q_{\mathsf{Activate}}(\lambda)$ be an upper bound on the number of Activate queries that an adversary in the PB-SK experiment makes. The adversary $\mathcal{C}$ takes a DDH tuple (g, X^*, Y^*, Z^*) as input and chooses $i, j \xleftarrow{R} \{1, \ldots, q_{\mathsf{Activate}}(\lambda)\}$. It then generates a key pair $(mpk, msk) \leftarrow \mathsf{KeyGen}_{\mathcal{S}}(1^\lambda)$ and runs $\mathcal{A}(msk)$. $\mathcal{C}$ responds to all of $\mathcal{A}$'s queries according to the rules of $\mathsf{Expt}^{\text{PB-SK}}_{\Pi_{\mathcal{S},\mathbb{G}},\mathcal{A}}(\lambda)$, except that it inserts the Diffie-Hellman values X^* and Y^* into the i^{th} and j^{th} sessions instead of generating a random group element. We refer to these session as s_i and s_j. If $\mathcal{A}$ queries $\mathsf{SKReveal}(s_i)$ or $\mathsf{SKReveal}(s_j)$, $\mathcal{C}$ aborts. When $\mathcal{A}$ queries $\mathsf{Test}(s_{u^*,\ell^*})$, $\mathcal{C}$ aborts *unless* $s_{u^*,\ell^*} = s_i$ and $s_{u',\ell'} = s_j$. Assuming it does not abort, $\mathcal{C}$ sets $k \leftarrow Z^*$. When $\mathcal{A}$ terminates and returns a guess b', $\mathcal{C}$ returns b' as its guess for the DDH problem.

Since the test session s_{u^*,ℓ^*} and its matching session $s_{u',\ell'}$ are chosen by the adversary $\mathcal{A}$ independently of the choice of i and j, $\Pr(\mathcal{C} \text{ does not abort}) \geq \frac{1}{q^2_{\mathsf{Activate}}}$. Whenever it does not abort, $\mathcal{C}$ wins the DDH game if and only if $\mathcal{A}$ wins the PB-SK experiment.

Combining results from Case 1 and Case 2, we see that

$$\begin{aligned}
\mathsf{Adv}^{\text{PB-SK}}_{\Pi_{\mathcal{S},\mathbb{G}},\mathcal{A}}(\lambda) &= \Pr(b' = b) = \Pr(b' = b|M)\Pr(M) + \Pr(b' = b|\neg M)\Pr(\neg M) \\
&\leq \frac{1}{q^2_{\mathsf{Activate}}(\lambda)}\mathsf{Adv}^{\text{DDH}}_{\mathbb{G},\mathcal{C}}(\lambda)\Pr(M) + \Pr(b' = b|\neg M)\mathsf{Adv}^{\text{PB-Forge}}_{\mathcal{S},\mathcal{B}}(\lambda) \\
&\leq \frac{1}{q^2_{\mathsf{Activate}}(\lambda)}\mathsf{Adv}^{\text{DDH}}_{\mathbb{G},\mathcal{C}}(\lambda) + \mathsf{Adv}^{\text{PB-Forge}}_{\mathcal{S},\mathcal{B}}(\lambda)
\end{aligned}$$

which is negligible as required. □

8 Conclusions

We have introduced the notion of predicate-based key exchange, given a security model, and presented a secure protocol satisfying the security definitions. Our security model for predicate-based key exchange can also be specialized to attribute-based key exchange, a cryptographic task for which there was previously no rigourous security definition.

Our security model incorporates two notions of security: session key security and credential privacy. We have argued that credential privacy is an essential property of predicate-based key exchange; without it, we might as well use certificates to link public keys and a list of credentials. However, achieving credential privacy requires careful consideration of the networking layer over which the protocol runs, as the addressing information of messages – the packet headers – may leak information. In practice, then, a secure deployment of predicate-based key exchange may rely on an anonymising network such as Tor.

The protocol we have presented is a generic protocol that combines any secure predicate-based signature scheme with a Diffie-Hellman construction, providing efficiency and simplicity.

Future work. The major security models for public-key-based authenticated key exchange have an additional query to allow revealing some of the session variables: either a SessionStateReveal query [CK01], which reveals the session state variables stored during the protocol, or an EphemeralKey-Reveal query [LLM07] which reveals all randomness used during the run of a protocol. Adding either of these queries to our security model would be a natural way to improve its security guarantees. Our generic protocol construction may still be secure with a SessionStateReveal query, but cannot be secure with an EphemeralKeyReveal query unless the underlying signature scheme is secure against revealing the randomness used in signing. No existing schemes have been shown to have this property, at least in the case of attribute-based or predicate-based signatures.

Our definition of credential privacy for predicate-based key exchange is computational in nature, but our proof for the generic construction relies on the perfect privacy of the underlying signature scheme, as defined by Maji et al. [MPR08]. However, it seems plausible that a suitably defined computational notion of credential privacy would suffice. It may also be possible to give alternative constructions based on ciphertext-policy predicate-based encryption schemes, though as yet only ciphertext-policy attribute-based encryption schemes exist.

Finally, predicate-based key exchange could be extended to support multiple, independent, mutually distrusting, potentially corrupt, credential authorities, as in multiple attribute authorities for attribute-based signature schemes [MPR08, §4].

Acknowledgements

The authors are grateful for discussions with Juanma González Nieto. J.B. was supported by Australian Research Council (ARC) Discovery Project DP0666065.

References

[AKB07] Guiseppe Ateniese, Jonathan Kirsch, and Marina Blanton. Secret handshakes with dynamic and fuzzy matching. In *Proc. Internet Society Network and Distributed System Security Symposium (NDSS) 2007*. Internet Society, 2007. URL http://www.isoc.org/isoc/conferences/ndss/07/papers/secret-handshakes.pdf.

[Bon98] Dan Boneh. The decision Diffie-Hellman problem. In Joe P. Buhler, editor, *Proc. Algorithmic Number Theory 3rd International Symposium (ANTS) '98*, *LNCS*, volume 1423, pp. 48–63. Springer, 1998. DOI:10.1007/BFb0054851. EPRINT http://crypto.stanford.edu/~dabo/abstracts/DDH.html.

[BR93] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *Advances in Cryptology – Proc. CRYPTO '93*, *LNCS*, volume 773, pp. 232–249. Springer,

1993. DOI:10.1007/3-540-48329-2_21. Full version available as URL http://www-cse.ucsd.edu/~mihir/papers/key-distribution.html.

[BW07] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In Salil Vadhan, editor, *Theory of Cryptography Conference (TCC) 2007*, *LNCS*, volume 4392, pp. 535–554. Springer, 2007. DOI:10.1007/978-3-540-70936-7_29.

[BWJM97] Simon Blake-Wilson, Don Johnson, and Alfred Menezes. Key agreement protocols and their security analysis. In Michael Darnell, editor, *Cryptography and Coding – 6th IMA International Conference*, *LNCS*, volume 1355. Springer, 1997. DOI:10.1007/BFb0024447.

[CCS07] Liqun Chen, Z. Cheng, and Nigel P. Smart. Identity-based key agreement protocols from pairings. *International Journal of Information Security*, **6**(4):213–241, July 2007. DOI:10.1007/s10207-006-0011-9. EPRINT http://eprint.iacr.org/2006/199.

[CK01] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In Birgit Pfitzmann, editor, *Advances in Cryptology – Proc. EUROCRYPT 2001*, *LNCS*, volume 2045, pp. 453–474. Springer, 2001. DOI:10.1007/3-540-44987-6_28. Full version available as EPRINT http://eprint.iacr.org/2001/040.

[CK03] Liqun Chen and Caroline Kudla. Identity based authenticated key agreement protocols from pairings. In *Proceedings 16th IEEE Computer Security Foundations Workshop (CSWF-16)*, pp. 219–233. IEEE, 2003. DOI:10.1109/CSFW.2003.1212715. Revised version appears as [CK04].

[CK04] Liqun Chen and Caroline Kudla. Identity based authenticated key agreement protocols from pairings, 2004. EPRINT http://eprint.iacr.org/2002/184. Revised version of [CK03].

[DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proc. 13th USENIX Security Symposium*. The USENIX Association, 2004. URL http://www.usenix.org/events/sec04/tech/dingledine.html.

[GBG10] Malakondayya Choudary Gorantla, Colin Boyd, and Juan González Nieto. Attribute-based authenticated key exchange. Unpublished manuscript, 2010.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Rebecca Wright, Sabrina De Capitani de Vimercati, and Vitaly Shmatikov, editors, *Proc. 13th ACM Conference on Computer and Communications Security (CCS)*, pp. 89–98. ACM, 2006. DOI:10.1145/1180405.1180418. Full version available as EPRINT http://eprint.iacr.org/2006/309.

[Gün89] Christoph G. Günther. An identity-based key-exchange protocol. In Jean-Jacques Quisquater and J. Vandewalle, editors, *Advances in Cryptology – Proc. EUROCRYPT '89*, *LNCS*, volume 434, pp. 29–37. Springer, 1989. DOI:10.1007/3-540-46885-4_5.

[KCD09] Dalia Khader, Liqun Chen, and James H. Davenport. Certificate-free attribute authentication. In Matthew G. Parker, editor, *Cryptography and Coding – 12th IMA International Conference*, *LNCS*, volume 5921, pp. 301–325. Springer, 2009. DOI:10.1007/978-3-642-10868-6_18.

[Kha07a] Dalia Khader. Attribute based group signature with revocation, 2007. EPRINT http://eprint.iacr.org/2007/241.

[Kha07b] Dalia Khader. Attribute based group signatures, 2007. EPRINT http://eprint.iacr.org/2007/159.

[Kha08] Dalia Khader. Authenticating with attributes, 2008. EPRINT http://eprint.iacr.org/2008/031.

[KP05] Caroline Kudla and Kenneth G. Paterson. Modular security proofs for key agreement protocols. In Bimal Roy, editor, *Advances in Cryptology – Proc. ASIACRYPT 2005*, *3788*, volume LNCS, pp. 549–565. Springer, 2005. DOI:10.1007/11593447_30.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel Smart, editor, *Advances in Cryptology – Proc. EUROCRYPT 2008*, *LNCS*, volume 4965, pp. 146–162. Springer, 2008. DOI:10.1007/978-3-540-78967-3_9. Full version available as EPRINT http://eprint.iacr.org/2007/404.

[LAS+10] Jin Li, Man Ho Au, Willy Susilo, Dongqing Xie, and Kui Ren. Attribute-based signature and its applications. In *Proc. 2010 ACM Symposium on Information, Computer and Communications Security (ASIACCS'10)*. ACM Press, 2010. URL http://www.ece.iit.edu/~kren/Attribute-based%20Signature%20and%20its%20Application.pdf. To appear.

[LK08] Jin Li and Kwangjo Kim. Attribute-based ring signatures, 2008. EPRINT http://eprint.iacr.org/2008/394.

[LLM07] Brian LaMacchia, Kristin Lauter, and Anton Mityagin. Stronger security of authenticated key exchange. In Willy Susilo, Joseph K. Liu, and Yi Mu, editors, *First International Conference on Provable Security (ProvSec) 2007*, *LNCS*, volume 4784, pp. 1–16. Springer, 2007. DOI:10.1007/978-3-540-75670-5_1.

[MPR08] Hemanta Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-based signatures: Achieving attribute-privacy and collusion-resistance, 2008. EPRINT http://eprint.iacr.org/2008/328.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology – Proc. CRYPTO '84*, *LNCS*, volume 196, pp. 47–53. Springer, 1984. DOI:10.1007/3-540-39568-7_5.

[SS09] Siamak F. Shahandashti and Reihaneh Safavi-Naini. Threshold attribute-based signatures and their application to anonymous credential systems. In Bart Preneel, editor, *Progress in Cryptology – AFRICACRYPT 2009*, *LNCS*, volume 5580, pp. 198–216. Springer, 2009. DOI:10.1007/978-3-642-02384-2_13. Full version available as EPRINT http://eprint.iacr.org/2009/126.

[SSW09] Emily Shen, Elaine Shi, and Brent Waters. Predicate privacy in encryption systems. In Omer Reingold, editor, *Theory of Cryptography Conference (TCC) 2009*, *LNCS*, volume 5444, pp. 457–473. Springer, 2009. DOI:10.1007/978-3-642-00457-5_27.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology – Proc. EUROCRYPT 2005*, *LNCS*, volume 3494, pp. 457–473. Springer, 2005. DOI:10.1007/11426639_27.

[WXB09] Hao Wang, Qiuliang Xu, and Tao Ban. A provably secure two-party attribute-based key agreement protocol. In *Proceedings of the 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 1042–1045, 2009. DOI:10.1109/IIH-MSP.2009.92.

[WXF09a] Hao Wang, Qiuliang Xu, and Xiu Fu. Revocable attribute-based key agreement protocol without random oracles. *Journal of Networks*, **4**(8):787–794, October 2009. DOI:10.4304/jnw.4.8.787-794.

[WXF09b] Hao Wang, Qiuliang Xu, and Xiu Fu. Two-party attribute-based key agreement protocol in the standard model. In Fei Yu, Jian Shu, and Guangxue Yue, editors, *Proceedings of the 2009 International Symposium on Information Processing (ISIP'09)*, pp. 325–328. Academy Publisher, 2009. URL http://www.academypublisher.com/proc/isip09/papers/isip09p325.pdf.