Secret Public Key Protocols Revisited

Hoon Wei Lim* and Kenneth G. Paterson**

Information Security Group
Royal Holloway, University of London Egham,
Surrey TW20 0EX, UK
{h.lim,kenny.paterson}@rhul.ac.uk

Abstract. Password-based protocols are important and popular means of providing human-to-machine authentication. The concept of secret public keys was proposed more than a decade ago as a means of securing password-based authentication protocols against off-line password guessing attacks, but was later found vulnerable to various attacks. In this paper, we revisit the concept and introduce the notion of identity-based secret public keys. Our new identity-based approach allows secret public keys to be constructed in a very natural way using arbitrary random strings, eliminating the structure found in, for example, RSA or ElGamal keys. We examine identity-based secret public key protocols and give informal security analyses, indicating that they are secure against off-line password guessing and other attacks.

1 Introduction

The use of *secret public keys* in password-based authentication protocols was first proposed by Gong *et al.* [19] in 1993. As implied by its name, a secret public key is a standard public key which can be generated by a user or a server, and is known only to themselves but is kept secret from a third party. A secret public key within a password-based protocol, when encrypted with a user's password, should serve as an unverifiable text[1]. This may significantly increase the difficulty of password guessing even if it is a poorly chosen password as an attacker has no way to verify if he has made the correct guess. The secret public key can then be used by the user for encrypting protocol messages. However, it may not be easy to achieve unverifiability of text by simply performing naive symmetric encryption on public key of standard types such as RSA or ElGamal. This was overlooked in [19] and other variants of secret public key protocols in [18,28], but later found to be the main culprit in various attacks on the protocols. These include undetectable on-line password guessing attacks from

* This author was supported by the EPSRC under grant EP/D051878/1.
** This author was supported by the European Commission under contract IST-2002-507932 (ECRYPT).

[1] Verifiable text/plaintext is a term popularised by Lomas *et al.* in [24]. It refers to a message that contains information that is recognisable when decrypted, whether or not it was predictable in advance.

B. Christianson et al. (Eds.): Security Protocols 2006, LNCS 5087, pp. 237–256, 2009.

Ding and Horster [17] and number theoretic attacks due to Patel [25]. It is worth noting that the attacks discovered in [17] may not work against a secret public key protocol which uses a secure public key encryption scheme such as RSA-OAEP [6]. Nevertheless, Patel's attacks seem to be one of the crucial factors that caused diversion of interest away from using secret public keys in password-based protocols. The concept of secret public keys, therefore, was thought to be unworkable. For example, in more recent work on password-based protocols that requires servers' public keys[2] (*e.g.* [11,20]), it is assumed that the public keys are fixed and known to all users.

Contributions. The aims of this paper are twofold: (i) we revisit the notion of secret public keys and uncover some unexplored potential benefits of using identity-based secret public keys, through identity-based cryptography (IBC), in password-based protocols; and (ii) we propose three-party and two-party identity-based secret public key protocols and their respective heuristic security analyses.

In our quest to revive the notion, we introduce some new properties for secret public keys. In the IBC setting, we show that an identity-based secret public key can offer more flexibility in terms of key distribution. For example, an identity-based secret public key can be computed by a user on-the-fly without needing his authentication server to transport the key to him. More importantly, a random string can be used as the identifier for constructing a secret public key. This technique can offer a clean and natural way of eliminating any predictable structure in the secret public key. Through this, the number theoretic attacks that plague existing secret public key protocols can easily be prevented.

Since both public and private keys in the IBC setting are kept secret, we also propose the notion of *secret signatures* which seem to provide data confidentiality in addition to their original cryptographic use, *i.e.* authentication and non-repudiation. This appears to provide additional properties in conventional secret public key protocols and in password-based authentication protocols in general.

Related Work. Extensive work on password-based key exchange protocols (which rely on user passwords only) has already been carried out. See for example [1,2,3,5,13,14,22], which all originate from [8,9]. In order to circumvent off-line password guessing attacks, Bellare *et al.* [5,7] proposed the use of a mask generation function $\mathcal{E}(\cdot)$ as an instantiation of the encryption primitive for encrypting a Diffie-Hellman component, rather than using a standard block (or stream) cipher. For instance, a user with his password PW can encrypt a Diffie-Hellman component g^x by calculating $g^x \cdot H(PW)$, where H is a hash function mapping onto the Diffie-Hellman group and which is modelled as a random oracle in security proofs. Thus the result of the encryption is a group element. This

[2] We classify password-based authentication protocols into two categories: (i) those which require the usage of the server's (or the user's) public key, and sometimes together with the user's password, as a key-encrypted key; and (ii) those which require the user's password only for key transport.

special encryption primitive, which needs to be carefully implemented, is crucial in preventing any information leakage about the password when an attacker mounts a guessing attack. To decrypt and recover g^x, one can simply divide the ciphertext by $H(PW)$. All recent work, such as [1,2,3], utilises this encryption primitive for their password-based key exchange protocols.

The use of algorithms from a public key encryption scheme in a secret key setting is not new. In 1978, Hellman and Pohlig [21] introduced the Pohlig-Hellman symmetric key cipher based on exponentiation. Two different keys are involved in the symmetric key cipher, namely, a secret encrypting key e for the sender and a secret decrypting key d for the receiver, where $e \neq d$. Obviously, the communicating parties must agree in advance to share these two symmetric keys. In more recent work, Brincat [15] investigated how shorter RSA public/private key pairs can be used securely in the secret key world. This is slightly different from [21], as each user has his own secret public/private key pair in [15]. Another related concept is that of public key privacy from Bellare *et al.* [4]. The notion of indistinguishability of keys in public key privacy is an extension of the ciphertext privacy concept: given a set of public keys and a ciphertext generated by using one of the keys, the adversary cannot tell which public key was used to generate the ciphertext. In this chapter, we will make use of identity-based (secret) public keys in the secret key setting. These public keys are known only to the senders and receivers, and thus indistinguishability of encryptions and keys somewhat similar to [4] can be achieved. Moreover, in such a setting, a signature can be made verifiable to only a specific recipient, hence the moniker *secret signature*. In many ways, the concepts of secret public key encryption and signatures seem to be closely related to the notion of signcryption with key privacy from Libert and Quisquater [23]. The proposal of [23] combined Zheng's work on signcryption [30] and the key privacy concept of [4]. Our concept of secret signatures is also related to, but different from, the strongest security notion for undeniable and confirmer signatures called invisibility in [16].

Organisation. The outline of the remainder of this paper is as follows. In Section 2, we review the first proposed secret public key protocol and highlight its problems. Section 3 briefly describes identity-based encryption and signature schemes that will be used in our new approach to secret public keys. In Section 4, we explain and discuss some new properties of secret public keys in the identity-based setting. In Section 5, we propose two variants of identity-based secret public key protocols. We also provide informal security analyses of the protocols. We conclude in Section 6.

2 Secret Public Key Protocols and Attacks

In this section, we revisit the first secret public key protocol proposed in the literature [19]. We will explain what the problems are with the protocol. This will motivate our introduction of identity-based techniques to this area.

Notation. We use $\hat{PK}$ and $\hat{SK}$ to represent a secret public key (SPK henceforth) and its matching private key, respectively. These are no different from conventional asymmetric keys except that they are *both* kept secret. PW denotes a password-derived symmetric key which is shared between a user and an authentication server. A nonce and a random number are represented by n and r, respectively. We use the notation $Enc_{\hat{PK}}(\cdot)$ to indicate asymmetric encryption using a secret public key $\hat{PK}$ and $\{\cdot\}_K$ for symmetric encryption under a symmetric key K. In the three-party scenarios that we will discuss in this section, we use A and B to denote two communicating parties, while S denotes a trusted authentication server whose role is to distribute a copy of a randomly generated session key to both A and B. Other notations will be introduced as they are needed.

The GLNS SPK Protocol. Gong *et al.* [19] envisaged that using secret public keys in a password-based protocol may be useful in a situation where the public keys are needed for certain protocol messages but the protocol participants do not know in advance the public key of their authentication server. In addition, they implicitly assumed that a secret public key could be viewed as a nonce which, when encrypted with a password, offers unverifiability of text. Assuming A and B share their respective passwords with the authentication server S, the server can distribute fresh copies of public keys to A and B encrypted using their respective passwords as symmetric keys at the beginning of each protocol run. Each public key is only known between the server and the relevant participant. This seems to make traditional chosen plaintext attacks more difficult, as the encryption keys are not known to the attacker. The details of the SPK protocol of [19] are depicted in Protocol 1.

Protocol 1. *The GLNS SPK Protocol*

(1). $A \rightarrow S : A,\ B$
(2). $S \rightarrow A : A,\ B,\ n_S,\ \{\hat{PK}_{SA}\}_{PW_A},\ \{\hat{PK}_{SB}\}_{PW_B}$
(3). $A \rightarrow B : Enc_{\hat{PK}_{SA}}(A, B, n_{A1}, n_{A2}, c_A, \{n_S\}_{PW_A}),\ n_S, r_A,\ \{\hat{PK}_{SB}\}_{PW_B}$
(4). $B \rightarrow S : Enc_{\hat{PK}_{SA}}(A, B, n_{A1}, n_{A2}, c_A, \{n_S\}_{PW_A}),$
$\qquad Enc_{\hat{PK}_{SB}}(B, A, n_{B1}, n_{B2}, c_B, \{n_S\}_{PW_B})$
(5). $S \rightarrow B : \{n_{A1},\ K_{AB} \oplus n_{A2}\}_{PW_A},\ \{n_{B1},\ K_{AB} \oplus n_{B2}\}_{PW_B}$
(6). $B \rightarrow A : \{n_{A1},\ K_{AB} \oplus n_{A2}\}_{PW_A},\ \{H(r_A),\ r_B\}_{K_{AB}}$
(7). $A \rightarrow B : \{H(r_B)\}_{K_{AB}}$

As shown in Protocol 1, S generates two new sets of secret public/private key pairs $(\hat{PK}_{SA}, \hat{SK}_{SA})$, $(\hat{PK}_{SB}, \hat{SK}_{SB})$ and distributes the public components to A in encrypted form whenever A initiates the protocol run. Here, c_A and c_B are sufficiently large random numbers known as confounders. They serve no purpose other than to confound guessing attacks based on some verifiable texts. Also, H is assumed to be a well-designed hash function.

In [19], the authors assumed that so long as the secret public keys $\hat{PK}_{SA}$ and $\hat{PK}_{SB}$ are randomly generated, it will be difficult for the attacker to verify if

his password guesses on $\{\hat{PK}_{SA}\}_{PW_A}$ or $\{\hat{PK}_{SB}\}_{PW_B}$ are correct. In reality, however, this is not completely true. When using conventional public keys such as RSA exponents or Diffie-Hellman components, the keys contain certain number theoretic structure even though they are randomly generated. This, in turn, may allow the attacker to verify his guessed passwords efficiently by predicting and checking the outcome of the decryption. For example, if $\hat{PK}_{SA}$ is an RSA public key of the form $N = pq$, then the attacker could expect the decryption of $\{\hat{PK}_{SA}\}_{PW_A}$ under a guess PW'_A for A's password to be an odd integer. This allows the elimination of half of all passwords in a simple off-line guessing attack. It is this observation that led to Patel's study on various number theoretic attacks on secret public key protocols [25]. It is also worth noting that $\{\cdot\}_K$ must not represent the action of an authenticated encryption algorithm as this would also leak information that could be used to verify the correctness or otherwise of password guesses.

Patel's Attacks. As we have just seen, it can be dangerous to transmit an RSA modulus in encrypted form in an SPK protocol. Even if the ciphertext contains only an RSA exponent, *e.g.* $\{e\}_{PW}$, there are various number theoretic attacks that would reveal the password PW. For example, the attacker could expect the decryption of $\{e\}_{PW}$ under a guess PW'_A to be an odd integer; an even result would eliminate PW'_A as a possible password. Thus, some countermeasures against these number theoretic attacks such as padding or randomisation of the RSA exponent are inevitably required.

Patel [25] showed that even when moduli N are sent in clear, and e are randomised and padded, there is still a lethal off-line guessing attack. Protocol 2 illustrates Patel's RSA version of the SPK protocol. We only show the first 3 out of 7 protocol messages as this is sufficient to describe Patel's attack.

Protocol 2. *The RSA SPK Protocol*

(1). $A \rightarrow S : A,\ B$
(2). $S \rightarrow A : A,\ B,\ n_S,\ \{e_{SA}\}_{PW_A},\ N_A,\ \{e_{SB}\}_{PW_B},\ N_B$
(3). $A \rightarrow B : Enc_{e_{SA}}(A,\ B,\ n_{A1},\ n_{A2},\ c_A,\ \{n_S\}_{PW_A}),\ n_S,\ r_A,\ \{e_{SB}\}_{PW_B}$
$\vdots$ $\vdots$

An attacker can impersonate S and block A's communication with the real authentication server to mount the following attack.

1. When the attacker E detects A is sending message (1) to S, he blocks S's response from reaching A. E intercepts message (2) and replaces N_A with his own N'_A whose prime factors he knows. Also, since E does not know PW_A, he simply replaces $\{e_{SA}\}_{PW_A}$ with a random string R_A.
2. A unwittingly decrypts R_A with her password-derived key PW_A and obtains e'_{SA} which A believes was generated by S. Subsequently in message (3), A forwards $Enc_{e'_{SA}}(A,\ B,\ \ldots)$ to B.
3. E intercepts message (3) and can now perform off-line password guessing on R_A. For each possible PW'_A, E decrypts R_A and retrieves a possible

value for e'_{SA}. Since E knows the prime factors of N'_A, he has no problem computing the decryption exponent d'_{SA} for each value of e'_{SA}. By decrypting $Enc_{e'_{SA}}(A,\ B,\ \ldots)$ with d'_{SA} and checking if the plaintext is of the form $(A,\ B,\ \ldots)$, E can test if PW'_A is the correct password.

It was pointed out in [25] that the above attack on the RSA-based SPK protocol is unavoidable unless all protocol participants use an agreed-upon RSA modulus, or unless the protocol is radically modified.

Even supposing a discrete logarithm based SPK protocol was used, and the ciphertext (which contains a secret public key) transmitted to A was then of the form $\{g^x\}_{PW}$, where g is a generator of a subgroup of $\mathbb{Z}_p^*$ of prime order q and x is a random integer, the password can still be discovered. If a naive encryption of elements in the subgroup is performed with a standard block (or stream) cipher, then there is an off-line password guessing attack. The attacker simply decrypts $\{g^x\}_{PW}$ with a guessed password and observes if the resulting plaintext is an element of the subgroup. If it was an incorrect guess, the likelihood that g^x is not an element of the subgroup is at least $(p-q)/p > 1/2$. This attack can only be prevented by ensuring that decryption of $\{g^x\}_{PW}$ with a guessed password PW' always results in an element of the subgroup. Furthermore, it is also essential that public parameters such as g, p and q have been agreed *a priori* among the users. More examples and discussion on this subject can be found in [25,27]. Notice that this kind of attack is prevented using mask generation functions of the type discussed in Section 1.

From the above descriptions of various number theoretic attacks, it should be evident that designing a SPK protocol can be difficult and not without some extra costs in ensuring the predictable number theoretic structure within public keys is eliminated. These observations are crucial for motivating our identity-based approach. We will show that the aforementioned problems can be prevented easily and naturally, using identity-based techniques.

3 Background on Identity-Based Cryptography

Identity-based cryptography (IBC) was first introduced by Shamir [26]. Recently, there has been an increased intensity in research on IBC. This was mainly due to the seminal discovery of a practical and secure identity-based encryption (IBE) scheme by Boneh and Franklin [10] in 2001. Their scheme uses pairings over elliptic curves.

Background on Pairings. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of order q for some large prime q, where $\mathbb{G}_1$ is an additive group and $\mathbb{G}_2$ denotes a related multiplicative group. A pairing in the context of IBC is a function $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ with the following properties.

– *Bilinear*: Given $P, Q, R \in \mathbb{G}_1$, we have

$$\hat{e}(P, Q + R) = \hat{e}(P, Q) \cdot \hat{e}(P, R) \text{ and } \hat{e}(P + Q, R) = \hat{e}(P, R) \cdot \hat{e}(Q, R).$$

Hence, for any $a, b \in \mathbb{Z}_q^*$, $\hat{e}(aP, bQ) = \hat{e}(abP, Q) = \hat{e}(P, abQ) = \hat{e}(aP, Q)^b = \hat{e}(P, Q)^{ab}$.
- *Non-degenerate*: There exists a $P \in \mathbb{G}_1$ such that $\hat{e}(P, P) \neq 1$.
- *Computable*: If $P, Q \in \mathbb{G}_1$, $\hat{e}(P, Q)$ can be efficiently computed.

For any $a \in \mathbb{Z}_q^*$ and $P \in \mathbb{G}_1$, we write aP as the scalar multiplication of group element P by integer a. Typically, $\mathbb{G}_1$ is obtained as a subgroup of the group of points on a suitable elliptic curve over a finite field, $\mathbb{G}_2$ is obtained from a related finite field, and $\hat{e}$ obtained from the Weil or Tate pairing on the curve.

In what follows, we briefly sketch the popular Boneh and Franklin IBE scheme and an identity-based signature (IBS) scheme with message recovery due to Zhang *et al.* These will be used in our identity-based SPK protocols.

3.1 The Boneh-Franklin Identity-Based Encryption Scheme

The following four algorithms underpin Boneh and Franklin's IBE scheme [10].

SETUP: Given a security parameter $k \in \mathbb{Z}^+$, the algorithm:
1. specifies two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q, and a pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$;
2. chooses an arbitrary generator $P \in \mathbb{G}_1$;
3. defines four cryptographic hash functions, $H_1 : \{0,1\}^* \to \mathbb{G}_1^*$, $H_2 : \mathbb{G}_1^* \to \{0,1\}^n$ for some n, $H_3 : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_q^*$, and $H_4 : \{0,1\}^n \to \{0,1\}^n$; and
4. picks a master secret $s \in \mathbb{Z}_q^*$ at random and computes the matching public component as sP.

The system or public parameters are $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, sP, H_1, H_2, H_3, H_4 \rangle$.

EXTRACT: This algorithm is run to extract a private key $sH_1(\mathrm{ID})$ when given an arbitrary identifier string $\mathrm{ID} \in \{0,1\}^*$.

ENCRYPT: To encrypt a message $m \in \{0,1\}^n$ under an identifier ID, the public key used is $Q_{\mathrm{ID}} = H_1(\mathrm{ID})$. The algorithm selects a random $z \in \{0,1\}^n$ and sets $r = H_3(z, m)$. The resulting cipertext is then set to be:

$$c = \langle U, V, W \rangle = \langle rP, z \oplus H_2(g^r), m \oplus H_4(z) \rangle,$$

where $g = \hat{e}(Q_{\mathrm{ID}}, sP) \in \mathbb{G}_2$.

DECRYPT: To decrypt a ciphertext $c = \langle U, V, W \rangle$ encrypted using the identifier ID, the private key used is $sQ_{\mathrm{ID}} \in \mathbb{G}_1^*$. If $U \notin \mathbb{G}_1^*$, reject the ciphertext. The plaintext m is then recovered by performing the following steps:
1. compute $V \oplus H_2(\hat{e}(sQ_{\mathrm{ID}}, U)) = z$;
2. compute $W \oplus H_4(z) = m$; and
3. set $r = H_3(z, m)$ and if $U \neq rP$, reject the ciphertext, otherwise accept m as the decryption of c.

It is a common assumption that the SETUP and EXTRACT algorithms are run by a trusted authority called the Private Key Generator (PKG) within a domain. All users within the domain are assumed to share the same system parameters.

We remark that the above IBE scheme is known to be secure against adaptive chosen ciphertext attacks (IND-ID-CCA) provided the Bilinear Diffie-Hellman problem is hard. This means that even though an adversary has access to some decryption keys associated to some identifiers (apart form the public key ID^* being attacked), he would still not able to deduce any useful information about an encrypted message using ID^* or the decryption key corresponding to ID^*. See [10] for further details.

3.2 The Zhang-Susilo-Mu Identity-Based Signature Scheme with Message Recovery

Using the same notation as above, we describe an IBS scheme with message recovery due to Zhang *et al.* [29].

SETUP**:** The PKG selects k_1 and k_2 such that $|q| = k_1 + k_2$. It also defines additional hash functions $H_0 : \{0,1\}^* \to \mathbb{Z}_q^*$, $F_1 : \{0,1\}^{k_2} \to \{0,1\}^{k_1}$, and $F_2 : \{0,1\}^{k_1} \to \{0,1\}^{k_2}$.

The system parameters are now $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, sP, H_0, H_1, F_1, F_2, k_1, k_2 \rangle$.

EXTRACT**:** As above.

SIGN**:** Given a private key sQ_{ID} and a message $m \in \{0,1\}^{k_2}$, the signer computes:

1. $v = \hat{e}(P,P)^k$, where $k \in \mathbb{Z}_q^*$;
2. $f = F_1(m) \| (F_2(F_1(m)) \oplus m)$;
3. $r = (H_1(v) + f) \bmod q$; and
4. $U = kP - r(sQ_{\mathrm{ID}})$.

The signature σ is (r, U). The length of the signature is $|r| + |U| = |q| + |\mathbb{G}_1|$.

VERIFY**:** Given a signature $\sigma = (r, U)$ signed by a user with a public key $Q_{\mathrm{ID}} = H_1(\mathrm{ID})$, the verifier computes

$$f = r - H_1(\hat{e}(U,P)\hat{e}(Q_{\mathrm{ID}}, sP)^r) \text{ and } m = [f]_{k_2} \oplus F_2([f]^{k_1}).$$

The verifier also checks if $[f]^{k_1} = F_1(m)$. The signature is accepted as valid if and only if this equation holds. Here $[f]^{k_1}$ denotes the left-most k_1 bits of the string f, while $[f]_{k_2}$ denotes the right-most k_2 bits of the string f.

The security of this scheme is based on the hardness of the computational Diffie-Hellman problem. To obtain approximately similar security as a standard 1024-bit RSA signature and a 2^{-80} probability of a successful forgery by an adversary, $|k_1| \leq 80$ is needed if a group element of $\mathbb{G}_1$ is represented by 171 bits [29]. The size of the message is limited to k_2, where $k_2 = |q| - k_1$.

4 New Properties from Identity-Based Secret Public Keys

We now present properties from identity-based SPKs by using the Boneh-Franklin IBE and the Zhang-Susilo-Mu IBS schemes. Pre-distribution or fixing of some

public/system parameters is common in password-based protocols. In this section and the next, for ease of exposition, we assume that the system parameters for the Boneh-Franklin IBE and the Zhang-Susilo-Mu IBS schemes can be distributed by the server to all its users during the user registration phase using an out-of-band mechanism. This is important as failure to use an authentic set of system parameters would allow the attacker to inject his own chosen parameters. Also, during the registration phase between a user and the server, the user will pick a password *pwd* and send an image PW of the password to the server. Typically, one might set $PW = H_0(pwd) \cdot P$, where $H_0 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, $\mathbb{G}_1$ is a group of prime order q used elsewhere in the protocol, and P generates $\mathbb{G}_1$. Note then that the server only knows PW and not *pwd*. The actual password *pwd* still remains private to the user only. In some cases where *pwd* and PW are used together, stronger authentication can be provided in the sense that the user's authenticity can still be guaranteed even if the string PW stored in the server is revealed. This technique of using an image of the actual user-selected password is common to many password-based protocols, for example [1,5,7,9].

Here, we present and discuss some interesting properties of identity-based SPKs (ID-SPKs henceforth) which are new as compared to conventional SPK protocols based on RSA or Diffie-Hellman. These properties can be obtained from using the Boneh-Franklin and Zhang-Susilo-Mu schemes, and they form the basis and motivation for the ID-SPK protocols that we will discuss in Section 5.

4.1 ID-SPK as Secret Identifier

In the conventional IBC setting, an identifier refers to some public information which represents a user and is known to all parties. Here, however, we work with secret identifiers, that is, identifiers only known to the user A (or B) and the server S. These can be obtained by binding a secret value such as a password to an identifier. Such an ID-SPK of the form $\hat{PK} = H_1(\texttt{user} \,\|\, \texttt{password} \,\|\, \texttt{policy})$ can be generated by both the user and the server on-the-fly. Here `policy` denotes constraints that can be included in the ID-SPK such as a date, nonces, or roles. In other words, the server does not need to distribute a fresh secret public key to its users, in contrast to [19,28]. Here we assume the users have access to the server's fixed system parameters. For example, referring back to Protocol 1, when A initiates the protocol she could, in principle, skip messages (1) & (2) and transmit message (3) to B as follows:

$$(3).\ A \rightarrow B : Enc_{\hat{PK}_{AS}}(A, B, \dots)$$

where $\hat{PK}_{AS} = H_1(A\|B\|S\|PW_A\|\text{“10102005”})$ denotes a public key in the IBE scheme of [10]. Here “10102005” represents a date. A date with more granularity (*e.g.* concatenated with time) or a nonce may well be needed to ensure freshness of $\hat{PK}_{AS}$. We remark that the Boneh-Franklin IBE scheme is probabilistic and thus the attacker cannot use a guessed password PW'_A to verify his guess by generating $Enc_{\hat{PK}_{AS}}(A, B, \dots)$ and comparing it with the actual ciphertext produced by A, even if he knows all the plaintext components.

On the server side, the server can extract the matching private key for $\hat{PK}_{AS}$ using its master secret. Unless the attacker can break the IBE scheme or recover the master secret, the above ciphertext is resistant to password guessing attacks. This identity-based technique offers a form of non-interactive distribution of secret public keys from the server to its users.

In the above example, A uses an ID-SPK encryption scheme which is adapted from the full version of the IBE scheme of [10] with the encryption key only known to the user and the server. Formal security definitions and proofs of security for ID-SPK encryption schemes are beyond the scope of this paper and will be addressed in our future work.

4.2 Random String as ID-SPK

We have explained earlier in Section 2 that a naive encyption of an RSA exponent or a group element with a standard block cipher would lead to effective off-line password guessing attacks. Therefore, some form of padding or randomisation of the keys is needed. In the IBC setting, we note that a random string with arbitrary length without any predictable structure can also be used as an identifier. The corresponding public key can be derived by hashing. Since now only a random string needs to be encrypted under the user password, the possibility of using a standard block cipher for the encryption is opened up[3]. For example, in Protocol 1, the server can transport random strings ST_A and ST_B to A and B, respectively, in message (2) as follows:

(2). $S \rightarrow A : A,\ B,\ n_S,\ \{ST_A\}_{PW_A},\ \{ST_B\}_{PW_B}$
(3). $A \rightarrow B : Enc_{\hat{PK}_{SA}}(A,\ B,\ n_{A1},\ n_{A2},\ n_S),\ n_S,\ r_A,\ \{ST_B\}_{PW_B}$
(4). $B \rightarrow S : Enc_{\hat{PK}_{SA}}(A,\ B,\ n_{A1},\ n_{A2},\ n_S), Enc_{\hat{PK}_{SB}}(B,\ A,\ n_{B1},\ n_{B2},\ n_S)$

Since ST_A and ST_B are just random strings, they do not contain any predictable structure which may leak some information to the attacker as in the case of RSA or Diffie-Hellman keys. Subsequently, users A and B can derive their ID-SPKs $\hat{PK}_{SA} = H_1(A\|B\|S\|ST_A)$ and $\hat{PK}_{SB} = H_1(B\|A\|S\|ST_B)$, respectively, and respond to the server via messages (3) and (4). If the server can decrypt B's reply and recover n_S from both the ciphertexts produced with $\hat{PK}_{SA}$ and $\hat{PK}_{SB}$, it can be assured that the users have received the correct random strings. Thus, A and B are authenticated to S. The use of random strings as identifiers is a key property from our identity-based approach which may give the concept of SPK protocols new life.

We remark that to prevent off-line attacks, ciphertexts obtained by encryption under the keys $\hat{PK}_{SA}$ and $\hat{PK}_{SB}$ must not leak useful information about ST_A and ST_B, respectively. This is not a traditional requirement of a public key encryption scheme (it is related to the public key privacy concept in [4]). Also note that since we use a probabilistic encryption scheme here, we have removed the use of confounders c_A and c_B originally proposed in Protocol 1 in messages

[3] However, it is still necessary to take care to avoid attacks based on the introduction of redundancy, for example padding, in the block cipher encryption.

(3) and (4). Furthermore, users A and B no longer need to encrypt n_S with their respective passwords in their replies to S, in messages (3) and (4). This is because users A and B can demonstrate their knowledge of respective passwords by their ability to construct correct keys from ST_A and ST_B.

4.3 Secret Signatures

In what follows, we show some extended properties that an ID-SPK can offer as compared to a conventional SPK. Again, referring to Protocol 1, if in the protocol A (and B) selects and sends ST_A (and ST_B) to the server (rather than the server sending it to the user), we can, in principle, remove messages (1) & (2) and modify messages (3) – (5) as follows:

$$\begin{aligned}
&(3).\ A \rightarrow B : Enc_{\hat{PK}_{SA1}}(A,\ B,\ ST_A),\ r_A \\
&(4).\ B \rightarrow S : Enc_{\hat{PK}_{SA1}}(A,\ B,\ ST_A),\ r_A, \\
&\qquad\qquad\quad Enc_{\hat{PK}_{SB1}}(A,\ B,\ ST_B),\ r_B \\
&(5).\ S \rightarrow B : Sig_{\hat{SK}_{SA2}}(K_{AB}),\ Sig_{\hat{SK}_{SB2}}(K_{AB})
\end{aligned}$$

Note that we have replaced nonces n_{A1}, n_{A2}, n_{B1} and n_{B2} in Protocol 1 by random strings ST_A and ST_B. For ease of exposition, we concentrate on the interaction between A and S. In message (3), A encrypts a random string ST_A with an ID-SPK $\hat{PK}_{SA1} = H_1(A\|B\|S\|PW_A)$. It is obvious that a symmetric encryption of the form $\{A,\ B, \ldots,\ ST_A\}_{PW_A}$ cannot be used in message (3) because the identities of A and B are verifiable texts. The server responds with a signature generated with a private key associated with the public key $\hat{PK}_{SA2} = H_1(S\|A\|B\|PW_A\|ST_A)$. The reason for doing this will be clear when we look at the motivation for using $Sig_{\hat{SK}}(\cdot)$, a signature scheme with a private key $\hat{SK}$, in message (5). As compared to the modification of Protocol 1 given in Section 4.2, the server cannot reply to A with an encrypted message using an ID-SPK constructed from $H_1(S\|A\|B\|PW_A\|ST_A)$. This is mainly because in such an asymmetric model (where the user only knows an easy-to-remember password and the server has access to the secret public/private key pairs), only the server itself can extract the corresponding private key. This prompts the requirement to use a secret signature which not only provides non-repudiation of the signed message and message recovery, but also preserves message confidentiality. This last property is needed because the server wants only A and B to be able to verify the signatures and recover the signed messages. This, in turn, leads us to the use of an ID-SPK signature scheme with message recovery which can be adapted from [29]. So long as the verification keys used in the scheme of [29] are kept secret between the intended parties, our concept of secret signatures can be used. However, we remark that the IBS scheme with message recovery must be used carefully because the scheme provides message integrity. In other words, a simple off-line password guessing attack would be enabled if a secret signature was created based on a private key corresponding to $\hat{PK}_{SA2} = H_1(S\|A\|B\|PW_A)$. For instance, the attacker could construct an ID-SPK $\hat{PK}'_{SA2} = H_1(S\|A\|B\|PW'_A)$ using a guessed password PW'_A and then

attempt to verify the signature. If he used the wrong password, the VERIFY algorithm would return an error message. Because of that, the identifier from which the verifying key is derived must contain a secret value chosen from a space much larger than the password space. We achieve this by including ST_A (or ST_B) in the identifier. It is also worth mentioning that a secret signature should not leak information about the signing key, the verifying key, or the plaintext that has been signed.

As we have explained earlier, a secret identifier can bind a user's password naturally to a secret public/private key pair. As such, secret signatures may be beneficial in a password-based protocol when one or both of the following conditions apply:

(i). Non-repudiation, confidentiality, and integrity of a signed message are required.
(ii). An additional line of defence is desirable (*e.g.* assuming the server keeps its master secret in a tamper-resistant hardware token or smartcard, the attacker cannot impersonate the server to any of its users even if the users' passwords are exposed).

Security definitions and proofs of security for ID-SPK signature schemes with message recovery will be addressed in our subsequent work on secret public keys.

5 The ID-SPK Protocols

In the previous sections, we learned that to exploit the advantages of using SPKs in a password-based protocol, the keys must not contain any predictable structure, such as that appearing in RSA or discrete logarithm-based systems. This section presents complete three-party and two-party ID-SPK protocols which can solve this structural issue in a clean and natural way. These protocols build on the ideas introduced in the previous section. We assume that all the protocol participants have agreed on some public/system parameters for the ID-SPK encryption and signature schemes *a priori*.

Before we look at the ID-SPK protocols, it may be useful to classify some common attacks on password-based protocols.

- *On-line password guessing attacks*: The attacker chooses a password from his dictionary and tries to impersonate a user. He verifies the correctness of his guess based on responses from the server. If the impersonation fails, the attacker tries again using a different password from his dictionary. Note that the attacker can also impersonate the server to the user by intercepting and modifying a message originating from the server before forwarding it to the user (assuming the server has used the user's password in some way in creating the message). He can then verify his password guesses based on responses from the user.
- *Off-line password guessing attacks*: The attacker records past communication and makes a verifiable guess using a password from his dictionary. If the

guess fails, the attacker tries again with a different password until the correct password is found. No on-line participation of a server (or a user) is required and the attacks take place without the knowledge of the actual protocol participants.
- *Attacks exploiting exposed secrets*: The attacker may occasionally have access to sensitive information such as past session keys or a user's password. This is possible when the user's machine or the server are compromised, or the user's password is revealed through a keystroke logger. It is a desirable security property that exposure of past session keys will not lead to the exposure of the user's password and vice versa.
- *Undetectable on-line password guessing attacks*: The attacker mounts an on-line guessing attack. However, a failed guess cannot be detected and logged by the server (or the user). In other words, the protocol participants cannot distinguish a genuine protocol message from a modified (malicious) message.

Security Model. We sketch here our definition of the security for a password-based ID-SPK protocol, using an informal security model. In the model, there is an adversary E, who is allowed to watch regular runs of the protocol between a user, $U \in \mathcal{U}$, where $\mathcal{U}$ is a set of protocol users, and a server S. E can actively communicate with the user and the server in replay, impersonation, and man-in-the-middle attacks. The adversary can prompt one of the parties to initiate new sessions. In each session, E can see all the messages sent between U and S. Furthermore, he can intercept the messages and modify or delete them. Also, E gets to see whether S accepts the authentication or not. In addition, we allow the adversary to establish as many "accounts" as he wishes with the server using his own chosen passwords. He can then run arbitrarily many authentication sessions using these accounts to obtain information for his attacks.

It is clear that if the user picks a password from his dictionary $\mathcal{D}$, then the adversary that attempts n active impersonation attacks (or on-line guessing attacks) over n distinct sessions with the server can succeed with probability at least $n/|\mathcal{D}|$ by trying a different password from $\mathcal{D}$ in each attempt.

Definition 1 (*Informal*). *We say that the ID-SPK protocol is secure if all the following conditions are satisfied.*

1. *No useful information about a session key is revealed to the adversary during a successful protocol run and the exposure of past session keys does not leak any information about the current session key.*
2. *The adversary cannot discover the correct user password after n active impersonation attempts with probability significantly higher than $n/|\mathcal{D}|$.*
3. *The protocol is resistant to off-line password guessing attacks.*
4. *The protocol is resistant to undetectable on-line password guessing attacks.*
5. *The exposure of the user's past session keys will not lead to the exposure of the user's password and vice versa.*

We remark that formal security model and definition, such as those used in [5], have not been employed in this paper. This is because the main objective of the paper is to explore new ways of using SPKs in the IBC setting.

5.1 The Three-Party ID-SPK Protocol

In [18], Gong further optimised the original SPK protocol in [19] by reducing the number of protocol messages to reduce the communication costs incurred by the protocol. We further modify Gong's optimised SPK protocol by building on the example given in Section 4.3, as shown in Protocol 3.

Protocol 3. *The Modified Gong SPK Protocol*

(1). $A \rightarrow B : A,\ r_A,\ Enc_{\hat{PK}_{A1}}(A,\ ST_A)$
(2). $B \rightarrow S : B,\ Enc_{\hat{PK}_{B1}}(B,\ ST_B),\ A,\ r_A,\ Enc_{\hat{PK}_{A1}}(A,\ ST_A)$
(3). $S \rightarrow B : Sig_{\hat{SK}_{B2}}(K_{AB}),\ Sig_{\hat{SK}_{A2}}(K_{AB})$
(4). $B \rightarrow A : Sig_{\hat{SK}_{A2}}(K_{AB}),\ \mathrm{MAC}_{F(K_{AB})}(B,\ A,\ r_A),\ r_B$
(5). $A \rightarrow B : \mathrm{MAC}_{F(K_{AB})}(A,\ B,\ r_B)$

In Protocol 3, users A and B select their respective random strings ST_A and ST_B and encrypt them with an ID-SPK. As before, $\hat{PK}_{A1} = H_1(A\|B\|S\|PW_A)$ and $\hat{PK}_{B1} = H_1(B\|A\|S\|PW_B)$. The server recovers ST_A and ST_B, and computes private keys $\hat{SK}_A$ and $\hat{SK}_B$ matching the ID-SPKs $\hat{PK}_{A2}$ and $\hat{PK}_{B2}$ constructed from the random strings: $\hat{PK}_{A2} = H_1(S\|A\|B\|PW_A\|ST_A)$ and $\hat{PK}_{B2} = H_1(S\|B\|A\|PW_B\|ST_B)$. The private keys are then used to sign a session key. We assume that an IBS scheme with message recovery is used, so that the intended recipients are able to recover the session key using their knowledge of the ID-SPKs. These secret signatures also provide non-repudiation. Even though this is rarely a requirement in protocols for authentication and key establishment, it automatically provides the important data integrity and data origin authentication services [12]. Note that $\mathrm{MAC}_{F(K_{AB})}(B,\ A,\ r_A)$ and $\mathrm{MAC}_{F(K_{AB})}(A,\ B,\ r_B)$ in messages (4) and (5) are used by A and B, respectively, to prove to each other that they are indeed sharing the same session key. This provides key confirmation. Here, F denotes a key derivation function.

To improve the performance of Protocol 3, $Sig_{\hat{SK}_{A2}}(K_{AB})$ and $Sig_{\hat{SK}_{B2}}(K_{AB})$ in message (3) can be replaced with $\{K_{AB}\}_{F(ST_A)}$ and $\{K_{AB}\}_{F(ST_B)}$, respectively.

Security Analysis. Protocol 3 shows that users A and B communicate with S using secret identifiers $\mathrm{ID}_A = A\|B\|S\|PW_A$ and $\mathrm{ID}_B = B\|A\|S\|PW_B$, respectively. These identifiers involve the users' passwords. Since S is the only party who has knowledge of PW_A and PW_B apart from A and B, the users should receive the same session key created by the server provided the correct private keys are used to transport the session key. If A and B can successfully recover K_{AB} from their respective received secret signatures, they can be assured of the authenticity of the server.

It is clear that requirement 1 of Definition 1 can be satisfied if the session key is randomly generated by the server. Moreover, the session key cannot be computed directly by the adversary E.

By observing a protocol run, E can gather information by intercepting the protocol messages, such as $Enc_{\hat{PK}_{A1}}(A,\ ST_A)$, $Enc_{\hat{PK}_{B1}}(B,\ ST_B)$, $Sig_{\hat{SK}_{A2}}(K_{AB})$

and $Sig_{\hat{SK}_{B2}}(K_{AB})$. However, since we assume that the ID-SPK encryption scheme used in this protocol is IND-ID-CCA secure, E cannot gain any useful information about ST_A and ST_B from $Enc_{\hat{PK}_{A1}}(A, ST_A)$ and $Enc_{\hat{PK}_{B1}}(B, ST_B)$ without knowledge of the master secret held by the server. As for the session key transportation in the form of secret signatures from the server to the users, E can choose his own verification keys in an attempt to recover the session key. However, there seems to be no efficient way for E to predict the correct ID-SPK if the ID-SPK signature scheme used in the protocol offers appropriate security. In particular, we assume that E cannot distinguish a secret signature from a randomly generated string if the identifier is constructed using sufficient randomness. We also assume that the adversary cannot forge valid secret signatures, impersonating the server to users. Apart from that, it is very unlikely that E can impersonate a legitimate user by guessing the user's password. This is so since the adversary's impersonation attack would be detected immediately by the server if the user's chosen random string cannot be recovered successfully from message (2). Note that the number of impersonation attempts can be kept acceptably small by using mechanisms that can log and control the number of failed authentication attempts. A brute force attack on message (3) or (4) to deduce the session key can be easily thwarted by using random strings ST_A and ST_B with entropy significantly larger than the password space of $\mathcal{D}$. Also, so long as ST_A and ST_B are fresh and randomly generated for each protocol run, E would not be able to mount a replay attack. It is thus conjectured that requirement 2 is satisfied.

When E uses a password $PW'_A \in \mathcal{D}$ to mount an off-line password guessing attack on a recorded $Enc_{\hat{PK}_{A1}}(A, ST_A)$, there is no way for the adversary to verify the correctness of $\hat{PK'}_{A1} = H_1(A\|B\|S\|PW'_A)$ if the ID-SPK encryption is randomised and IND-ID-CCA secure. If E selects $PW'_A \in \mathcal{D}$ and ST'_A at random, computes $\hat{PK'}_{A2} = H_1(S\|A\|B\|PW'_A\|ST'_A)$, and then attempts to verify $Sig_{\hat{SK}_{A2}}(K_{AB})$, his check will almost certainly fail since the entropy of ST_A is much larger than the entropy of PW_A. Thus this form of off-line guessing attack will not succeed and therefore, Protocol 3 also satisfies requirement 3.

If E has a valid account with S, he may possibly mount an insider attack by impersonating A to S, pretending to be wanting to establish a session key K_{AE} with himself. In the attack, E initiates the protocol by computing $Enc_{\hat{PK'}_{A1}}(A, ST'_A)$ with a guessed password PW'_A, and hence $\hat{PK'}_{A1} = H_1(A\|B\|S\|PW'_A)$. However, once this message has reached S, the server should get an error message when decrypting $Enc_{\hat{PK'}_{A1}}(A, ST'_A)$ using the decryption key matching $\hat{PK'}_{A1}$. Therefore it is clear that the protocol can detect on-line guessing attacks and thus requirement 4 is satisfied.

On certain rare occasions, E may have access to A's or B's machine and thus the past session keys shared between them are exposed. However, since E has no knowledge of the master secret of S and the matching private component of $\hat{PK}_{A2}$, E still cannot determine PW_A even though he can mount a brute-force attack on $\hat{PK}_{A2}$. On the other hand, if for some reason, E has the correct password for A, he may attempt to find the value of ST_A given A's password and

the ciphertext $Enc_{\hat{PK}_{A1}}(A,\, ST_A)$. Since the encryption scheme is IND-ID-CCA secure, E only has a negligible success probability to discover the correct ST_A. Also, since the value of the verification key for $Sig_{\hat{SK}_{A2}}(K_{AB})$ depends on the secret value ST_A, E can only recover the session key with negligible probability and forward secrecy of the protocol is preserved. Hence, requirement 5 is also satisfied. We conclude that Protocol 3 is a secure ID-SPK assuming that the ID-SPK encryption and signature schemes are appropriately secure.

Nevertheless, it is worth noting that if the server's master secret is compromised, the adversary can deduce the users' passwords without much difficulty. For instance, for each candidate password PW'_A, E can extract the private key matching the identifier $\mathrm{ID}'_A = A\|B\|S\|PW'_A$ and use it to attempt to decrypt $Enc_{\hat{PK}_{A1}}(A,\, ST_A)$ from message (1), and check if the decryption unveils A's identity. Hence, it is of the utmost importance that the server's master secret is kept private, for example by using a strong protective mechanism such as storing it in a tamper-resistant device.

5.2 The Two-Party ID-SPK Protocol

We now present a Diffie-Hellman type two-party ID-SPK protocol. Our protocol is adapted from [1,5] which make use of an encrypted Diffie-Hellman ephemeral key exchange. We apply the identity-based techniques that we introduced in Section 4 to obtain Protocol 4, as shown below.

Protocol 4. *The Diffie-Hellman ID-SPK Protocol*

(1). $A \rightarrow S : A,\ Enc_{\hat{PK}_{A1}}(aP)$
(2). $S \rightarrow A : S,\ Sig_{\hat{SK}_{A2}}(xP)$

In Protocol 4, the user randomly selects $a \in \mathbb{Z}_q^*$ and computes aP, where $P \in \mathbb{G}_1$ is part of the system parameters. A then encrypts the Diffie-Hellman component with $\hat{PK}_{A1} = H_1(A\|S\|PW_A)$ and sends message (1) to S. The server extracts the matching private key $\hat{SK}_{A1}$ with its master secret to recover aP. Subsequently, S picks a random number $x \in \mathbb{Z}_q^*$ and calculates xP. The server then extracts another private key which is associated with $\hat{PK}_{A2} = H_1(S\|A\|PW_A\|aP)$, produces $Sig_{\hat{SK}_{A2}}(xP)$, and transmits it to A. After receiving message (2), the user retrieves xP with $\hat{PK}_{A2}$. Both the user and the server calculate a session key as $K_{AS} = F(A\|S\|PW_A\|aP\|xP\|axP)$, where F is a key derivation function. Note that key confirmation can be provided by adding a third message from A to S, in which A provides a MAC computed on all the protocol messages using the session key (derived using a different key derivation function to F).

Security Analysis. As with Protocol 3, user A uses an ID-SPK, but in this case to transport a Diffie-Hellman ephemeral key aP to the server. It is worth noting that message (1) can be replayed but this is not an issue because the purpose of the protocol is to authenticate the session key. If the adversary E

has captured message (1) and replays it, he will not gain any information about the session key, unless he has access to a and to xP in message (2). Also, we note that since only S other than A has access to PW_A, S is authenticated to A when A successfully recovers xP using $\hat{PK}_{A2} = H_1(S\|A\|PW_A\|aP)$ (recall that an ID-SPK signature scheme provides a message integrity check).

Clearly, requirement 1 of Definition 1 can be satisfied if the ephemeral Diffie-Hellman components from A and S are randomly generated and information used to compute the session key including a, aP, x, xP, and PW_A cannot be computed directly by E.

E has access to $Enc_{\hat{PK}_{A1}}(aP)$ and $Sig_{\hat{SK}_{A2}}(xP)$ through watching a protocol run between A and S. However, since we assume that the ID-SPK encryption scheme used in this protocol is IND-ID-CCA secure, E cannot obtain any useful information about aP from $Enc_{\hat{PK}_{A1}}(aP)$ without knowledge of the master secret held by the server. Also, we assume that the ID-SPK signature scheme used in the protocol produces secret signatures $Sig_{\hat{SK}_{A2}}(xP)$ that are indistinguishable from random strings. Hence it is hard for E to deduce any information about the Diffie-Hellman component chosen by the server. Using analysis similar to that we used when discussing Protocol 3, it appears unlikely that E will successfully impersonate A in n attempts with probability significantly higher than $n/|\mathcal{D}|$ or mount a replay attack, provided aP and xP are fresh and their entropy is significantly higher than the entropy of $\mathcal{D}$. Also, the use of an incorrect password in generating $\hat{PK}_{A1}$ can be easily detected by the server when the server uses the wrong matching private key to recover aP. It is thus conjectured that requirements 2, 3 and 4 are satisfied.

It is possible that E may have access to A's machine and recover the past session keys used by A. In that case, despite the fact that E knows K'_{AS}, he must be able to reverse the key derivation function F in order to deduce A's password. On the other hand, if for some reason A's password is revealed to E, E may attempt to find the value of aP given A's password and the ciphertext $Enc_{\hat{PK}_{A1}}(aP)$. Since the encryption scheme is IND-ID-CCA secure, E only has a negligible success probability to find the correct aP. Also, since the value of the verification key for $Sig_{\hat{SK}_{A2}}(xP)$ depends on the secret value aP, E can only recover the session key with negligible probability. This is related to the forward secrecy of protocols discussed in [1,5]. Therefore, requirement 5 is also satisfied. We note that in addition to having met this requirement, even if E knows aP and xP, he has to solve the intractable CDH problem in order to calculate axP and hence the session key. We conclude that Protocol 4 is a secure ID-SPK protocol.

As with the security of Protocol 3, it is essential to have the server's master secret adequately protected to ensure that the aforementioned security conditions hold.

6 Conclusions

We studied the history of secret public key protocols and discussed some known problems with these protocols. We then explored some interesting properties

of identity-based cryptography which form the basis of our proposed identity-based secret public key protocols. These properties also allow us to convert a conventional identity-based encryption scheme and a standard identity-based signature scheme (with message recovery) into their secret public key equivalents.

We presented three-party and two-party identity-based secret public key protocols for key exchange. Our heuristic security analyses show that the protocols appear to be secure against off-line password guessing attacks and undetectable on-line password guessing attacks, and provide forward secrecy. The security definitions and proofs of the ID-SPK encryption and signature schemes, as well as formal security analyses of the proposed ID-SPK protocols in this paper, will be addressed in our further work on this subject.

Acknowledgements

The authors would like to thank Qiang Tang for his very helpful feedback on an earlier draft of this paper, and Kim-Kwang Raymond Choo for pointing out a flaw in an earlier version of Protocol 3.

References

1. Abdalla, M., Chevassut, O., Pointcheval, D.: One-time verifier-based encrypted key exchange. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 47–64. Springer, Heidelberg (2005)
2. Abdalla, M., Fouque, P., Pointcheval, D.: Password-based authenticated key exchange in the three-party setting. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 65–84. Springer, Heidelberg (2005)
3. Abdalla, M., Pointcheval, D.: Simple password-based encrypted key exchange protocols. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 191–208. Springer, Heidelberg (2005)
4. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 566–582. Springer, Heidelberg (2001)
5. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated key exchange secure against dictionary attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000)
6. Bellare, M., Rogaway, P.: Optimal asymmetric encryption – how to encrypt with RSA. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995)
7. Bellare, M., Rogaway, P.: The AuthA Protocol for Password-Based Authenticated Key Exchange. Contribution to IEEE P1363 (March 2000)
8. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: Proceedings of the 1992 IEEE Symposium on Security and Privacy, pp. 72–84. IEEE Computer Society Press, Los Alamitos (1992)
9. Bellovin, S.M., Merritt, M.: Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise. In: Proceedings of the 1st ACM Computer and Communications Security Conference, pp. 244–250. ACM Press, New York (1993)

10. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
11. Boyarsky, M.K.: Public-key cryptography and password protocols: The multi-user case. In: Proceedings of the 6th ACM Computer and Communications Security Conference, pp. 63–72. ACM Press, New York (1999)
12. Boyd, C., Mathuria, A.: Protocols for Authentication and Key Establishment. Springer, Berlin (2003)
13. Boyko, V., MacKenzie, P., Patel, S.: Provably secure password authenticated key exchange using Diffie-Hellman. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 156–171. Springer, Heidelberg (2000)
14. Bresson, E., Chevassut, O., Pointcheval, D.: Security proofs for an efficient password-based key exchange. In: Proceedings of the 10th ACM Computer and Communications Security Conference, pp. 241–250. ACM Press, New York (2003)
15. Brincat, K.: On the use of RSA as a secret key cryptosystem. Designs, Codes, and Cryptography 22(3), 317–329 (2001)
16. Chaum, D., van Heijst, E., Pfitzmann, B.: Cryptographically strong undeniable signatures, unconditionally secure for the signer. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 470–484. Springer, Heidelberg (1992)
17. Ding, Y., Horster, P.: Undetectable on-line password guessing attacks. ACM Operating Systems Review 29(4), 77–86 (1995)
18. Gong, L.: Optimal authentication protocols resistant to password guessing attacks. In: Proceedings of 8th IEEE Computer Security Foundations Workshop (CSFW 1995), pp. 24–29. IEEE Computer Society Press, Los Alamitos (1995)
19. Gong, L., Lomas, T.M.A., Needham, R.M., Saltzer, J.H.: Protecting poorly chosen secrets from guessing attacks. IEEE Journal on Selected Areas in Communications 11(5), 648–656 (1993)
20. Halevi, S., Krawczyk, H.: Public-key cryptography and password protocols. ACM Transactions on Information and System Security 2(3), 230–268 (1999)
21. Hellman, M.E., Pohlig, S.C.: Exponentiation Cryptographic Apparatus and Method. U.S. Patent #4,424,414, January 3 (1984) (expired)
22. Jablon, D.P.: Strong password-only authenticated key exchange. ACM SIGCOMM Computer Communication Review 26(5), 5–26 (1996)
23. Libert, B., Quisquater, J.-J.: Efficient signcryption with key privacy from gap Diffie-Hellman groups. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 187–200. Springer, Heidelberg (2004)
24. Lomas, T.M.A., Gong, L., Saltzer, J.H., Needham, R.M.: Reducing risks from poorly chosen keys. ACM Operating Systems Review 23(5), 14–18 (1989)
25. Patel, S.: Number theoretic attacks on secure password schemes. In: Proceedings of the 1997 IEEE Symposium on Security and Privacy, pp. 236–247. IEEE Computer Society Press, Los Alamitos (1997)
26. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
27. Steiner, M., Buhler, P., Eirich, T., Waidner, M.: Secure password-based cipher suite for TLS. ACM Transactions on Information and System Security 4(2), 134–157 (2001)
28. Tsudik, G., Herreweghen, E.V.: Some remarks on protecting weak keys and poorly chosen secrets from guessing attacks. In: Proceedings of the 12th IEEE Symposium on Reliable Distributed Systems (SRDS 1993), pp. 136–141. IEEE Computer Society Press, Los Alamitos (1993)

29. Zhang, F., Susilo, W., Mu, Y.: Identity-based partial message recovery signatures (or how to shorten ID-based signatures). In: Patrick, A.S., Yung, M. (eds.) FC 2005. LNCS, vol. 3570, pp. 45–56. Springer, Heidelberg (2005)
30. Zheng, Y.: Digital signcryption or how to achieve cost (Signature & encryption) << cost(Signature) + cost(Encryption). In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165–179. Springer, Heidelberg (1997)