Generalized Digital Certificate for User Authentication and Key Establishment for Secure Communications

Lein Harn and Jian Ren, *Senior Member, IEEE*

***Abstract*—Public-key digital certificate has been widely used in public-key infrastructure (PKI) to provide user public key authentication. However, the public-key digital certificate itself cannot be used as a security factor to authenticate user. In this paper, we propose *the concept of generalized digital certificate (GDC)* that can be used to provide user authentication and key agreement. A GDC contains user's public information, such as the information of user's digital driver's license, the information of a digital birth certificate, etc., and a digital signature of the public information signed by a trusted certificate authority (CA). However, the GDC does not contain any user's public key. Since the user does not have any private and public key pair, key management in using GDC is much simpler than using public-key digital certificate. The digital signature of the GDC is used as a secret token of each user that will never be revealed to any verifier. Instead, the owner proves to the verifier that he has the knowledge of the signature by responding to the verifier's challenge. Based on this concept, we propose both discrete logarithm (DL)-based and integer factoring (IF)-based protocols that can achieve user authentication and secret key establishment.**

***Index Terms*—Public-key digital certificate, user authentication, key management.**

I. Introduction

A digital certificate is the combination of a statement and a digital signature of the statement. The well-known digital certificate is the "X.509 public-key digital certificate" [1]. The statement generally contains the user's public key as well as some other information. The signer of the digital signature is normally a trusted certificate authority (CA). The X.509 public-key digital certificate has been widely used in public-key infrastructure (PKI) to provide authentication on the user's public key contained in the certificate. The user is authenticated if he is able to prove that he has the knowledge of the private key corresponding to the public key specified in the X.509 public-key digital certificate. However, the public-key digital certificate itself cannot be used to authenticate a user since a public-key digital certificate contains only public information and can be easily recorded and played back once it has been revealed to a verifier.

In this paper, we propose an innovative approach which enables a user to be authenticated and a shared secret session key be established with his communication partner using any general form of digital certificates, such as a digital driver's license, a digital birth certificate or a digital ID, etc. We call this kind of digital certificate as a *generalized digital certificate (GDC)*. A GDC contains user's public information and a digital signature of this public information signed by a trusted CA. However, in GDC, the public information does not contain any user's public key. Since user does not have any private and public key pair, this type of digital certificate is much easier to manage than the X.509 public-key digital certificates. The digital signature of the GDC is used as a secret token of each user. The owner of a GDC never reveals signature of GDC to a verifier in plaintext. Instead, the owner computes a response to the verifier's challenge to prove that he has the knowledge of the digital signature. Thus, owning a GDC can provide user authentication in a digital world. In addition, a secret session key can be established between the verifier and the certificate owner during this interaction.

There are three entities in a digital certificate application. They are the following:

a) Certificate Authority (CA): CA is the person or organization that digitally signs a statement with its private key. In PKI applications, the X.509 public-key digital certificate contains a statement, including the user's public key, and a digital signature of the statement. The difference between the GDC and the existing public-key digital certificate is that in a GDC, the public information does not contain any user's public key.

b) Owner of a GDC: The owner of the GDC is the person who receives the GDC from a trusted CA over a secure channel. The owner needs to compute a valid "answer" in response to the verifier's challenged "question" in order to be authenticated and establish a secret session key.

c) Verifier: The verifier is the person who challenges the owner of a GDC and validates the answer using the owner's public information and CA's public key.

In most paper-world user identification applications, a trusted authority is responsible for issuing identification card with user information, such as user name and a personal photo on the card, to each user. Each user can be successfully identified if the user owns a legitimate "paper certificate" and matches the photo on the card. The built-in tamper-resistant technology made the identification cards very difficult to be

Manuscript received October 27, 2010; revised March 25, 2011; accepted March 27, 2011. The associate editor coordinating the review of this paper and approving it for publication was W. Lou.

L. Harn is with the Department of Computer Science and Electrical Engineering, University of Missouri–Kansas City, MO 64110, USA (e-mail: harnl@umkc.edu).

J. Ren is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA (e-mail: renjian@egr.msu.edu).

This work was supported in part by the US National Science Foundation under CAREER Award CNS-0845812 and grants CNS-0848569 and CNS-1050326.

Digital Object Identifier 10.1109/TWC.2011.042211.101913

forged. Therefore, owning a paper certificate is the factor in the authentication process. In this paper, our goal is to propose a similar solution in electronic-world applications. We call it the *generalized digital certificate* (GDC). A GDC contains public information of the user and a digital signature of the public information signed by a trusted certificate authority. The digital signature will never be revealed to the verifier. Therefore, the digital signature of a GDC becomes a security factor that can be used for user authentication.

The rest of this paper is organized as follows. In the next section, we provide an overview of the related work. In Section III, we introduce some preliminaries and also describe discrete Logarithm (DL)-based user authentication and key establishment protocol using GDC. In Section IV, we describe integer factoring (IF)-based user authentication and key establishment protocol. We conclude in Section V.

II. Related Work

User authentication and key establishment are two fundamental services in secure communications. Extensive research has been conducted in both areas. However, unlike the GDC as we propose in this paper, most schemes in literature rely on the public-key digital certificates in providing user authentication and key establishment [2]–[4].

A traditional digital signature provides authentication of a given message to the receiver. However, this approach can sometimes violate the signer's privacy. A malicious receiver can reveal the sender's digital signature to any third-party without the sender's consent. Subsequently, anyone can access the signer's public key and validate the digital signature. In 1989, Chaum and Antwerpen [5] introduced the notion of an undeniable signature, which enables the signer to have a complete control over his/her signature. The verification of an undeniable signature requires participation of the message signer. However, this arrangement can prevent undesirable verifiers from validating the signature. The real problem of the undeniable signature is that the signer needs to authenticate the verifier before helping the verifier to validate the undeniable signature. Some recent works can be found in [6], [7].

Designated verifier signature (DVS) was first introduced in [8], and also in [9] independently, both in 1996. A DVS provides authentication of a given message to a specified verifier. One unique property of a DVS is that a valid DVS can be generated by the "real" signer or by the designated verifier. With this unique property, a DVS is different from a traditional digital signature in two aspects. (i) Since the designated verifier knows that he/she did not generate the DVS him/herself, the designated verifier is therefore convinced that the DVS was generated by the real signer. However, unlike the traditional digital signature, which can be verified by any verifiers, for the DVS, no third party member can determine the real signer of the DVS even with knowledge of the private key. (ii) A DVS provides authentication of a given message without non-repudiation property of the traditional digital signature. A DVS can replace the traditional digital signature in most applications and provide services with deniability.

In [8], a DVS scheme based on a non-interactive undeniable signature scheme with a trap-door commitment was proposed, however, this scheme is computationally inefficient. A DVS can be established by setting the number of signers in a ring signature to two, as proposed in [10], [11]. However, a DVS based on ring signatures does not provide strong designated verifier properties. In [12], a DL-based DVS scheme based on the combination of Schnorr signature [13] and Zheng signature [14] was proposed. It is a pairing-based variant of [10]. More recently, DVS schemes based on any bilinear map was proposed [15].

The concept of universal DVS (UDVS) was proposed in [16]. A UDVS is an ordinary digital signature with the additional functionality that allows the owner of a digital signature to convert the signature into a DVS of any designated verifier at his choice. The construction of a UDVS scheme (DVSBM) was based on a bilinear map. Three new UDVS constructions based on Schnorr [13] and RSA signatures [17] were proposed in [18]. Also, the ElGamal-based UDVS has been proposed in [15]. Some other related research on the DVS and UDVS can be found in [19]–[21].

Similar to our proposed scheme, there are three entities in each UDVS application: the CA, the owner of a digital signature, and the designated verifier. However, in a UDVS, the owner needs to convert the digital signature into a DVS non-interactively in order to authenticate a message. While in our proposed scheme, the owner of a digital certificate interacts with a verifier in order to prove the knowledge of the digital certificate and to be authenticated by the verifier.

Our proposed scheme is closely related to the ID-based cryptography [22]. In an ID-based cryptographic algorithms, each user needs to register at a private key generator (PKG) and identify himself before joining the network. Once a user is accepted, the PKG will generate a private key for the user. The user's identity (e.g. user's name or email address) becomes the corresponding public key. In this way, in order to verify a digital signature of a message, the sender sends an encrypted message to a receiver, a user only needs to know the "identity" of his communication partner and the public key of the PKG, which is extremely useful in cases like wireless communication where pre-distribution of authenticated public keys is infeasible. However, in an ID-based cryptographic algorithm, it is assumed that each user already knows the identity of his communication partner. Based on this assumption, there is no need, nor have feasible ways, to authenticate the identity. This is the main advantage of ID-based cryptography. Due to this assumption, ID-based cryptography is only limited to applications that communication entities know each other prior to communication. While in our proposed GDC scheme, the user does not need to know any information of his/her communication partner. The public information of a GDC, such as user's identity, can be transmitted and verified by each communication entity. Furthermore, this information is used to authenticate each other. In other words, our proposed schemes support general PKI applications, such as Internet e-commence, that communication entities do not need to know each other prior to the communication. Our proposed solution is based on the combination of a conventional digital signature scheme and the well-known (generalized) Diffie-Hellman assumption [23], [24].

III. DL-Based Protocol

A. Preliminaries

A paper certificate can be used as an user's authentication factor, but a public-key digital certificate cannot be used as an authentication factor in network applications. This is because a paper certificate cannot be easily forged or duplicated, but a public-key digital certificate can be easily recorded and played back.

In our scheme, the owner of a GDC never needs to reveal the digital signature of the GDC in plaintext to the verifier. Instead, the owner proves that he has knowledge of the digital signature by responding to the verifier's challenge. The knowledge of the digital signature on the GDC can provide user authentication. The proposed protocol should satisfy the following security requirements.

1) **Unforgeability**: A valid response can only be generated by the certificate owner who knows the digital signature of the GDC.
2) **One-wayness**: No other person can derive the digital signature of the certificate based on the interaction.
3) **Nontransferrability**: A response to a verifier's challenge cannot be transferred into a response to another verifier's challenge, which would otherwise create impersonation of the user.

Our proposed protocol is built on the combination of the traditional DL-based digital signature and the Diffie-Hellman Assumption (DHA) [23].

B. Review of ElGamal Digital Signature

In the ElGamal scheme [25], a large prime p and a generator g in the order of $p-1$ are assumed to be shared by all users. The signer selects a random private key $x \in [1, p-2]$ and computes the corresponding public key $y = g^x \bmod p$.

The signer first randomly selects a secret parameter $k \in [1, p-1]$ with $\gcd(k, p-1) = 1$ and computes $r = g^k \bmod p$. Then, s is solved by knowing the signer's secrets, x and k, as

$$m = ks + rx \bmod p - 1, \tag{1}$$

where m represents the message digest of the message m'. (r, s) is defined as the digital signature of the message m'. The signature (r, s) can be verified by checking whether the equation

$$g^m = y^r r^s \bmod p, \tag{2}$$

holds true.

In an ElGamal signature scheme, the parameter r of the signature can be computed off-line as $r = g^k \bmod p$. The signature component s is computed on-line. Readers can refer to [26] for more discussion on the design of DL-based signature schemes. Without loss of generality, we can represent the generalized signing equation for all DL-based signature schemes as $ax = bk + c \bmod p - 1$ where (a, b, c) are three parameters from the set of values (m, r, s). More specifically, each parameter can be a mathematical combination of (m, r, s). For example, the parameter a can be m, r or s. The verification equation is determined accordingly as $y^a = r^b g^c \bmod p$. There are 18 generalized ElGamal-type signature variants [26].

In the following discussion, we use the original ElGamal signature as an example to present our proposed protocol.

C. Diffie-Hellman Assumption (DHA)

Assume A and B have their private keys, x_A and x_B, and their corresponding public keys, $y_A = g^{x_A} \bmod p$ and $y_B = g^{x_B} \bmod p$, respectively, where p is a large prime integer and g is a primitive element of the multiplicative group modulo p. Only A and B can compute a shared secret $K_{A,B} = y_B^{x_A} = y_A^{x_B} = K_{B,A} \bmod p$.

DHA refers to the assumption that it is computationally infeasible to determine $K_{A,B}$ without knowing the private key x_A or x_B. However, solving the private key x_A or x_B from the corresponding public key y_A or y_B is equivalent to solving the discrete logarithm problem.

D. User Authentication and Key Establishment Protocol

1) Registration at CA: Let A be the certificate owner and B be the verifier. A needs to register at a CA to obtain a GDC. The CA generates an ElGamal signature (r_A, s_A) for user A's statement m'_A according to equation (1), where m_A is the message digest of the statement m'_A. Since the signature component r_A is a random integer and does not depend on m_A, it does not need to be kept secret. However, the signature component s_A is a function of the statement. Each owner needs to keep it secret from the verifier in the authentication process. Our user authentication and key establishment protocol is illustrated in Fig. 1.

2) Protocol: The authentication and key establishment protocol contains the following four steps:

1) The user A passes his user information m'_A and parameters (r_A, S_A) to the verifier B, where $S_A = r_A^{s_A} \bmod p$.
2) After receiving m'_A and (r_A, S_A), the verifier checks whether
$$g^{m_A} = y^{r_A} S_A \bmod p, \tag{3}$$
where y is the public key of the CA. If this equality holds true, the verifier B first randomly selects an integer $v_B \in [1, p-2]$, then computes a challenge $c_B = r_A^{v_B} \bmod p$ and send c_B to the user A. Otherwise, the user authentication fails and the protocol is stopped.
3) The user A first uses his secret s_A to compute the Diffie-Hellman secret key $K_{A,B} = c_B^{s_A} \bmod p$, $K'_{A,B} = D(K_{A,B})$, where $D(K_{A,B})$ represents a key derivation procedure with $K_{A,B}$ as an input. Then user A randomly selects an integer $v_A \in [1, p-2]$, computes $c_A = r_A^{v_A} \bmod p$ and the response $Ack = h(K'_{A,B}, c_B \| c_A)$, where $h(K'_{A,B}, c_B \| c_A)$ represents a one-way keyed-hash function under the key $K'_{A,B}$. The user A sends Ack and c_A back to B.
4) After receiving the Ack and c_A from the user A, the verifier B uses his secret v_B to compute the Diffie-Hellman shared secret key $K_{B,A} = S_A^{v_B} \bmod p$, $K'_{B,A} = D(K_{B,A})$, and checks whether $h(K'_{B,A}, c_B \| c_A) = Ack$ is true. If this verification is successful, the certificate owner A is authenticated by the verifier B and a one-time secret session key $c_B^{v_A} = r_A^{v_A v_B} = c_A^{v_B} \bmod p$ is

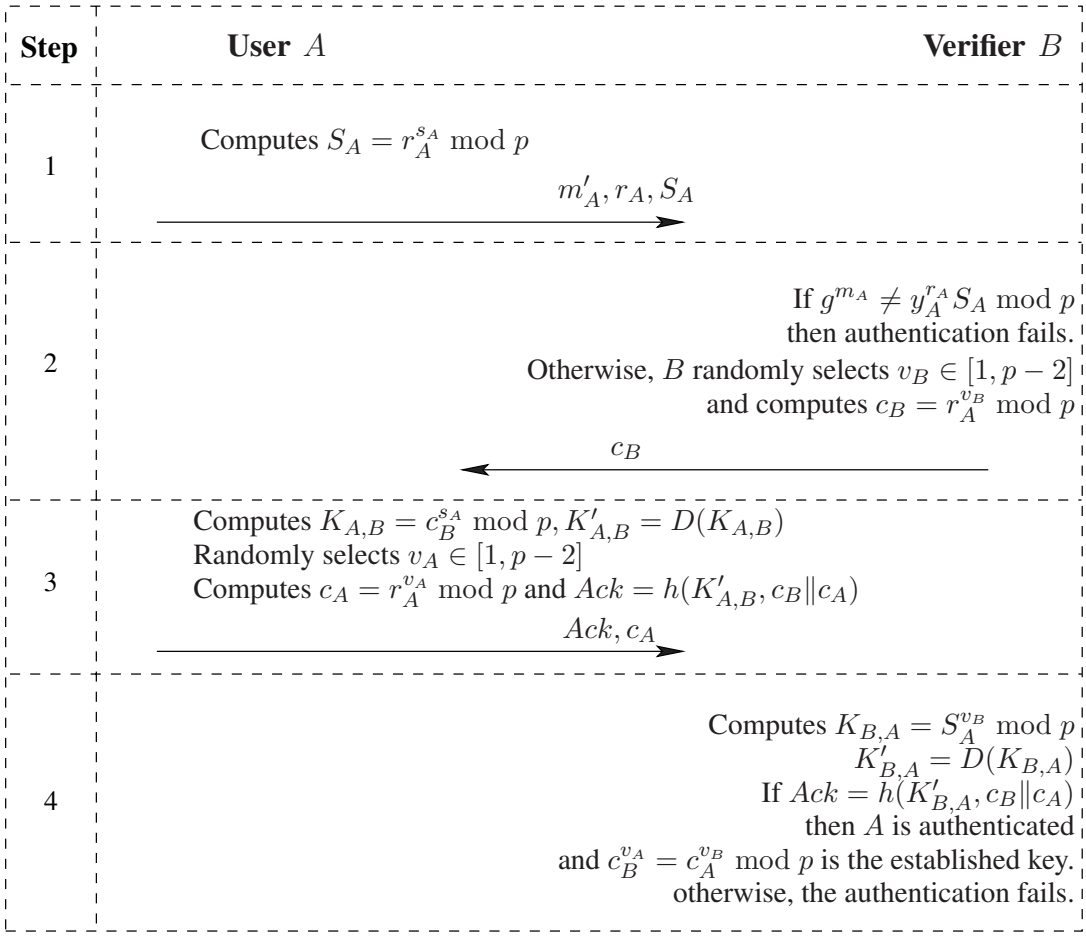


Fig. 1. DL-based authentication and key agreement protocol.

shared between A and B. This shared key can provide perfect forward security.

In order to be authenticated successfully by the verifier, in our protocol, the certificate owner needs to compute and send a valid pair (r_A, S_A) and Ack to the verifier in steps 1) and 3). The parameters (r_A, S_A) need to satisfy

$$g^{m_A} = y^{r_A} S_A \bmod p.$$

This pair of integers can be easily solved by anyone. However, we want to show that only the certificate owner A who knows the secret exponent of S_A can compute a valid Ack. This is because the verifier B can compute the one-time secret key $K_{B,A}$ used in generating the Ack as $K_{B,A} = S_A^{v_B} = r_A^{s_A v_B} \bmod p$. According to the DHA, the certificate owner A who knows the secret exponent of S_A can also compute $K_{A,B}$ as $K_{A,B} = c_B^{s_A} = r_A^{s_A v_B} = K_{B,A} \bmod p$. Thus, the certificate owner can interact with the verifier and be authenticated successfully.

Remark 1: As we have discussed previously, a valid S_A can be solved by anyone, including the verifier. Thus, technically, S_A does not need to be transmitted in step 2). However, if the prover sends S_A in step 2), it can help the verifier to terminate the protocol immediately once an invalid S_A is detected.

E. Security Analysis and Discussion

In this section, we will analyze the security of the proposed user authentication and key establishment protocol for the unforgeability, one-wayness and nontransferability.

a) Unforgeability: In order to perform a forgery attack, the attacker needs to present a valid pair (r_A, S_A) in step 1) and the corresponding Ack in step 3) in order to impersonate the certificate owner successfully. A valid pair (r_A, S_A) alone in step 1) cannot be used to authenticate the certificate owner since this pair of parameters can be solved easily by the attacker from equation (3). However, it is computationally infeasible for the attacker to find the discrete logarithm of S_A because the security of the ElGamal signature scheme. Therefore, it is computationally infeasible for the attacker to get a pair (r_A, s_A) to satisfy $g^{m_A} = y^{r_A} r_A^{s_A} \bmod p$. Due to the DHA, without knowing the secret exponent of S_A, it would be infeasible for the attacker to compute $K_{A,B}$ and forge a valid Ack in step 3). On the other hand, the certificate owner obtains the secret exponent of S_A from CA during the registration and the certificate owner can be authenticated successfully in step 3). In summary, the security of the unforgeability of our proposed protocol is provided through combination of the security of the ElGamal signature scheme and the DHA.

Therefore, the proposed user authentication and key establishment protocol is secure against forgery attacks.

b) One-wayness: In step 1), the certificate owner presents S_A to the verifier. The computation of secret s_A from S_A is infeasible since computation of s_A from the S_A is a discrete logarithm problem. Also, in step 3), the certificate owner uses the secret s_A to compute the Diffie-Hellman key $K_{A,B}$. Although the verifier knows the Diffie-Hellman key $K_{A,B}$; but due to the DHA, the verifier cannot obtain the secret s_A. Therefore, our proposed protocol satisfies the one-wayness property.

c) Nontransferability : Due to the DHA, a valid response Ack can only be generated by a certificate owner who knows the secret digital signature component s_A such that $r_A^{s_A} = S_A \bmod p$, or by a verifier who knows the random secret of a random challenge selected by the verifier. As the verifier selects a random challenge each time, the response is only valid for a one-time authentication.

Since the digital signature of a GDC is never passed to the verifier, the verifier cannot pass the complete GDC to any third party. There is no privacy intrusion problem in our protocol. Therefore, a valid response Ack cannot be transferred into a response of another verifier's challenge.

Our protocol enables a certificate owner to be authenticated and two one-time shared secret keys $K_{A,B}$ and $c_B^{v_A} = r_A^{v_A v_B} = c_A^{v_B} \bmod p$ be established between A, the certificate owner, who knows s_A such that $r_A^{s_A} = S_A \bmod p$, and the verifier B through the authentication protocol. The former is used to generate the Ack, and the latter is established shared secret key between A and B. In addition, it enables the owner to send a confirmation Ack to the verifier. Since the Diffie-Hellman secret shared key can be generated by either A or B, the certificate owner A can deny participating in the protocol.

Remark 2: In the original DHA, it is assumed that the generator g is a primitive element of the multiplicative group modulo p; while the parameter $r_A = g^k \bmod p$ in Theorem 1 is not necessarily a generator. However, we can ensure that r_A is a primitive element of the multiplicative group modulo p by requiring $gcd(k, p-1) = 1$ [27]. Particularly, when $p = 2p'+1$ is a safe prime, where p' is also a prime, we can ensure r_A is a primitive element of the multiplicative group modulo p if k is an odd number.

Remark 3: Similar to the ID-based cryptographic algorithms, our proposed protocol also has the key escrow problem, that is the CA knows the one-time secret session key shared between the users. Some cryptographic algorithms have been proposed to solve the key escrow problem of the ID-based signature (IBS) while enjoying the benefits of the IBS, such as certificateless digital signature (CDS) [28].

IV. IF-Based Protocol

In this section, we propose an IF-based user authentication and key establishment protocol. It is a combination of an on-line/off-line digital signature [29] and a generalized Diffie-Hellman assumption (GDHA) [24].

A. Review of On-line/Off-line Digital Signature

We will review the trapdoor hash families and the on-line/off-line signature scheme based on the trapdoor hash families.

A trapdoor hash family, introduced in [30] and formally defined in [29], consists of a pair $(\mathcal{I}, \mathcal{H})$, where $\mathcal{I}$ is a probabilistic polynomial-time key generation algorithm, and $\mathcal{H}$ is a family of randomized hash family. $\mathcal{I}$ generates a pair (HK, TK), where HK is a (public) hash key, and TK is its associated (private) trapdoor key. A trapdoor hash function in h is a hash function with a trapdoor secret. It is represented as $h_{HK}(m, s)$, where m is a message and s is an auxiliary random number. A trapdoor hash function must satisfy the following three requirements:

- **Efficiency**: Given a hash key HK and a pair (m, s), $h_{HK}(m, s)$ is computable in polynomial time.
- **Collision resistance**: There is no probabilistic polynomial-time algorithm A, on input HK, that can generate two pairs (m_1, s_1) and (m_2, s_2) such that $m_1 \neq m_2$ and $h_{HK}(m_1, s_1) = h_{HK}(m_2, s_2)$ with non-negligible probability.
- **Trapdoor collision**: Given pairs (HK, TK), (m_1, s_1) and an additional message m_2, there exists a probabilistic polynomial-time algorithm that generates s_2 such that
  - $h_{HK}(m_1, s_1) = h_{HK}(m_2, s_2)$.
  - If s_1 is uniformly distributed in $\mathcal{S}$, then the distribution of s_2 is computationally indistinguishable from uniform distribution in $\mathcal{S}$.

B. Factoring-Based Trapdoor Hash Function

Choose at random two safe primes p and q (i.e. primes such that $p' = (p-1)/2$ and $q' = (q-1)/2$ are primes) and compute $n = pq$. Choose at random an element g of order $\lambda(n)$, where $\lambda(n) = \text{lcm}(p-1, q-1) = 2p'q'$. The public hash key HK is (n, g) and the private trapdoor key TK is (p, q). The trapdoor hash function $h_{HK}(m, s)$ is defined as follows:

$$h_{HK}(m, s) = g^{m \| s} \bmod n, \tag{4}$$

where $\|$ denotes concatenation. To show that the $h_{HK}(m, s)$ is a trapdoor hash function under the factoring assumption, one needs to show that it fulfills the three main properties of a trapdoor hash function. The proof that $h_{HK}(m, s)$ is a factoring based trapdoor hash function can be found in [29].

For given pairs (HK, TK), (m_1, s_1) and an additional message m_2, to compute a trapdoor collision, we need to compute an s_2 such that $h_{HK}(m_1, s_1) = h_{HK}(m_2, s_2)$. According to equation (4), equivalently, we should have $g^{m_1 \| s_1} = g^{m_2 \| s_2} \bmod n$. That is, we need to find an s_2 such that $2^k m_1 + s_1 = 2^k m_2 + s_2 \bmod \lambda(n)$, where k is the size of the auxiliary parameter s. Given the trapdoor key $TK = (p, q)$, $\lambda(n)$ can be computed in polynomial time and hence s_2 can be computed in polynomial time by solving the linear equation

$$s_2 = 2^k (m_1 - m_2) + s_1 \bmod \lambda(n).$$

C. Signature scheme

In [29], a hash-sign-switch paradigm in which any regular digital signature scheme combined with a trapdoor hash family in $(\mathcal{I}, \mathcal{H})$ can be converted into an on-line/off-line signature scheme. Basically, in the off-line phase, a signer generates a

hash value to commit to an arbitrarily selected message. In the on-line phase, given a message, the signer finds a collision of the trapdoor hash to the previously calculated hash value. The collision point and the signature generated in the off-line phase can be presented as the signature for the message generated in the on-line phase.

Let $h_{HK}(m,s)$ be a trapdoor hash function, HK be the hash key, TK be the associated trapdoor key, VK be the verification key, and SK be the signing key for any regular digital signature scheme. The following describes the on-line/off-line signature scheme:

- Key generation algorithm GEN: Generate a pair (SK, VK) using a public-key generation algorithm and a pair (HK, TK) using the algorithm $\mathcal{I}$. The signing key is (SK, HK, TK) and the verification key is (VK, HK).
- Signing algorithm SIGN: Given a signing key (SK, HK, TK), the signing algorithm operates as follows:
  - Off-line phase: The signer randomly selects (m, s) and computes $h_{HK}(m,s)$, then uses his secret key SK to sign $h_{HK}(m,s)$ and obtain $\langle S_{SK}(h_{HK}(m,s))\rangle$. The signer stores m, s, $S_{SK}(h_{HK}(m,s))$ and optionally $h_{HK}(m,s)$ to avoid re-computation during the on-line phase.
  - On-line phase: Given a message m', the signer finds a collision of the trapdoor hash for (m,s) such that $h_{HK}(m',s') = h_{HK}(m,s)$. The signature of message m' is defined as $\langle S_{SK}(h_{HK}(m,s)), s', h_{HK}(m,s)\rangle$.
- Verification algorithm VERF: First verify $\langle S_{SK}(h_{HK}(m,s))\rangle$ using VK and $h_{HK}(m,s)$, and then compute $h_{HK}(m',s')$ to verify if $h_{HK}(m,s) = h_{HK}(m',s')$.

D. Generalized Diffie-Hellman Assumption (GDHA)

Assume A and B have their private keys x_A and x_B, and their corresponding public keys $y_A = g^{x_A} \bmod n$ and $y_B = g^{x_B} \bmod n$, respectively. Let $n = pq$, where p and q are two large primes. Then it is assumed that only A and B can compute a shared secret $K_{A,B} = y_A^{x_B} = y_B^{x_A} \bmod n$. GDHA refers to the assumption that it is computationally infeasible to determine $K_{A,B}$ without knowing the private key x_A or x_B. It has been shown in [24] that GDHA is a valid assumption as long as factoring Blum-integers is hard.

E. User Authentication and Key Establishment Protocol

1) Registration at CA: Let A be the certificate owner and B be the verifier. A needs to register at a CA to obtain a GDC. The CA generates an on-line/off-line digital signature, $(S_{SK}(h_{HK}(m',s')),\ s_A, h_{HK}(m',s'))$, for user A's statement m_A. Each owner needs to keep the signature s_A secret from the verifier in the authentication protocol. While proving knowledge of the secret component to the verifier, the owner conceals the secret component to the verifier during the authentication phase following the GDHA. Our user authentication and key establishment protocol is illustrated in Fig. 2.

2) Protocol: The authentication and key establishment protocol contains the following four steps:

1) The user A passes his user information m_A and parameters $(S_{SK}(h_{HK}(m',s')), S_A, h_{HK}(m',s'))$, to the verifier B, where $S_A = g^{s_A} \bmod n$.
2) After receiving m_A and $(S_{SK}(h_{HK}(m',s')), S_A, h_{HK}(m',s'))$, B first verifies that $(S_{SK}(h_{HK}(m',s'))$ is the signature of $h(m',s')$ using the VK. Then, computes
$$h_{HK}(m_A, S_A) = g^{2^k m_A} S_A \bmod n,$$
and verify if $h_{HK}(m_A, S_A) = h_{HK}(m',s')$, where k is the length of the secret exponent s_A. If this equality holds true, the verifier B first randomly selects an integer $v_B \in [1, n-1]$, then computes $c_B = g^{v_B} \bmod n$ and sends c_B to the user A. Otherwise, the user authentication fails and the protocol is stopped.
3) The user A first uses his secret s_A to compute the Diffie-Hellman secret key $K_{A,B} = c_B^{s_A} \bmod n$, $K'_{A,B} = D(K_{A,B})$. Then user A randomly selects an integer $v_A \in [1, n-1]$, computes $c_A = g^{v_A} \bmod n$ and the response $Ack = h(K'_{A,B}, c_B \| c_A)$, where $D(K_{A,B})$ represents a key derivation procedure with $K_{A,B}$ as an input, $h(K'_{A,B}, c_B \| c_A)$ represents a one-way keyed-hash function under the key $K'_{A,B}$. The user A sends Ack and c_A back to B.
4) After receiving the Ack and c_A from the user A, the verifier B uses his secret v_B to compute the Diffie-Hellman shared secret key $K_{B,A} = S_A^{v_B} \bmod n$, $K'_{B,A} = D(K_{B,A})$, and checks whether $h(K'_{A,B}, c_B \| c_A) = Ack$ is true. If this verification is successful, the certificate owner A is authenticated by the verifier B and a one-time secret session key $c_B^{v_A} = g^{v_A v_B} = c_A^{v_B} \bmod n$ is shared between A and B. This key can provide perfect forward security.

In order to be authenticated successfully by the verifier, in our protocol, the certificate owner needs to compute and sends valid parameters $(S_{SK}(h_{HK}(m',s')), S_A, h_{HK}(m',s'))$ and Ack to the verifier in steps 1) and 3). The parameter S_A needs to satisfy
$$h_{HK}(m',s') = g^{2^k m_A} S_A \bmod n.$$
This parameter can be easily solved by anyone or is publicly available. However, we want to show that only the certificate owner A who knows the secret exponent of S_A can compute a valid Ack. This is because the verifier B can compute the one-time secret key $K_{B,A}$ used in generating Ack as $K_{B,A} = S_A^{v_B} = g^{s_A v_B} \bmod n$. According to the GDHA, the certificate owner A who knows the secret exponent of S_A can also compute $K_{A,B}$ as $K_{A,B} = c_B^{s_A} = g^{s_A v_B} = K_{B,A} \bmod n$. Thus, the certificate owner can interact with the verifier and be authenticated successfully.

Remark 4: In our proposed protocol, CA generates an on-line/off-line digital signature for each registered user. The CA does not actually need the trapdoor property of the one-way hash function. In fact, The CA does not need the trapdoor key. *It only needs to use the one-way hash property to compute a hash value* S_A. Also, in order to construct an IF-based

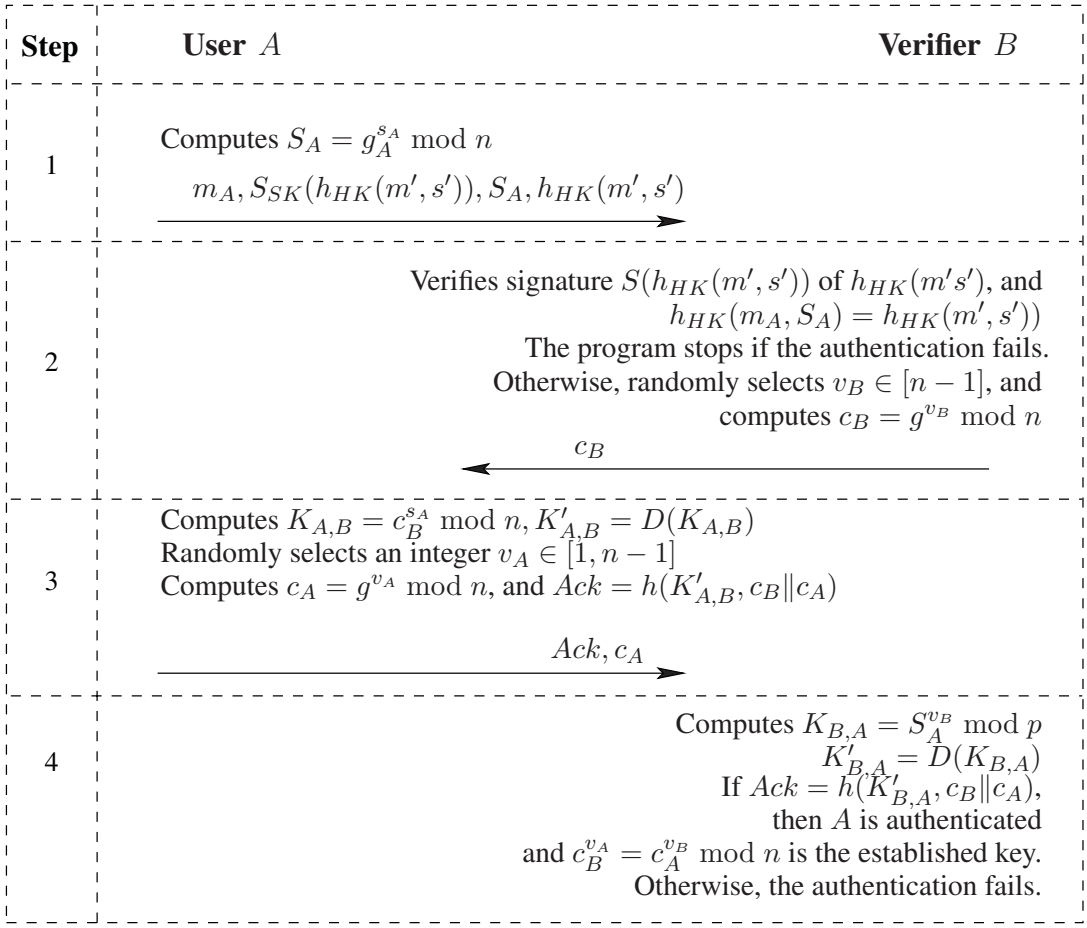


Fig. 2. IF-based authentication and key agreement protocol.

protocol, the CA needs to use the RSA signature to digitally sign the hash value $h(m', s')$.

F. Security Analysis and Discussion

The security of this protocol relies on the combination of the security of the RSA signature, collision resistance of the one-way hash function and the GDHA. The On-line/Off-line digital signature is secure against adaptive-chosen message attacks, provided that the original scheme is secure against generic chosen-message attacks [29]. It has also proved that the trapdoor hash function is collision resistance [29]. Similar to the security analysis presented in Section III-E for the DL-based protocol, the proposed IF-based protocol also satisfies the properties of unforgeability, one-wayness and nontransferability. The protocol also provides deniable authentication and protects privacy of the digital certificate.

V. Conclusion

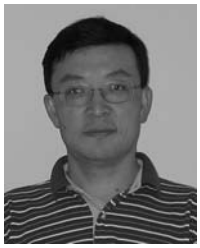
In this paper, we have proposed a novel design in using a GDC for user authentication and key establishment. In our design, a GDC does not contain the user's public key. Since the user does not have any private and public key pair, this type of digital certificate is much easier to manage than the X.509 public-key digital certificates. Our approach can be applied to both DL-based and IF-based public-key cryptosystems.

References

[1] Network Working Group, "Internet X.509 public key infrastructure certificate and crl profile, RFC: 2459," Jan. 1999.
[2] C. Tang and D. Wu, "An efficient mobile authentication scheme for wireless networks," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 1408-1416, Apr. 2008.
[3] G. Yang, Q. Huang, D. Wong, and X. Deng, "An efficient mobile authentication scheme for wireless networks," *IEEE Trans. Wireless Commun.*, vol. 9, pp. 168-174, Jan. 2010.
[4] J. Chun, J. Hwang, and D. Lee, "A note on leakage-resilient authenticated key exchange," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 2274-2279, May 2009.
[5] D. Chaum and H. van Antwerpen, "Undeniable signatures," *Advances in Cryptology - Crypto'89*, Lecture Notes in Computer Science, vol. 435, pp. 212-217, 1989.
[6] M. Bohøj and M. Kjeldsen, "Cryptography report: undeniable signature schemes," Tech. Rep., Dec. 15, 2006.
[7] X. Huang, Y. Mu, W. Susilo, and W. Wu, "Provably secure pairing-based convertible undeniable signature with short signature length," *Pairing-Based Cryptography -C Pairing 2007*, vol. 4575/2007 of *Lecture Notes in Computer Science*, pp. 367-391, Springer Berlin / Heidelberg, 2007.
[8] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications," *Advances in Cryptology - EUROCRYPT*, pp. 143-154, 1996. LNCS Vol 1070.
[9] D. Chaum, "Private signature and proof systems," 1996.
[10] R. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," *Advances in Cryptology-ASIACRYPT*, Lecture Notes in Computer Science, vol. 2248/2001, Springer Berlin / Heidelberg, 2001.
[11] J. Ren and L. Harn, "Generalized ring signatures," *IEEE Trans. Dependable Secure Comput.*, vol. 5, no. 4, Oct.-Dec., pp. 155-163, 2008.
[12] S. Saeednia, S. Kremer, and O. Markowitch, "An efficient strong designated verifier signature scheme," *ICISC 2003*, vol. 2836 of *Springer Lecture Notes in Computer Science*, pp. 40-54, 2003.

[13] C. Schnorr, "Efficient signature generation by smart cards," *J. Cryptology*, vol. 4, no. 3, pp. 161-174, 1991.
[14] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption << cost (signature) + cost (encryption)," *Advances in Cryptology - Crypto'97*, Lecture Notes in Computer Science vol. 1294, pp. 165-179, 1997.
[15] F. Laguillaumie and D. Vergnaud, "Designated verifier signatures: anonymity and efficient construction from any bilinear map." IACR e-print.
[16] R. Steinfeld, L. Bull, H. Wang, and J. Pieprzyk, "Universal designated verifier signatures," in *Asiacrypt'03*, vol. LNCS 2894, pp. 523-542, 2003.
[17] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. Assoc. Comp. Mach.*, vol. 21, no. 2, pp. 120-126, 1978.
[18] R. Steinfeld, H. Wang, and J. Pieprzyk, "Efficient extension of standard schnorr/rsa signatures into universal designated-verifier signatures," *PKC'04*, vol. Springer Lecture Notes in Computer Science of *2947*, pp. 86-100, 2004.
[19] H. Lipmaa, G. Wang, and F. Bao, "Designated verifier signature schemes: Attacks, new security notions and a new construction," in *ICALP'05*, 2005.
[20] Y. Li, W. Susilo, Y. Mu, and D. Pei, "Designated verifier signature: Definition, framework and new constructions," *Ubiquitous Intelligence and Computing*, vol. 4611/2007, Springer Berlin / Heidelberg, 2007.
[21] A. Mihara and K. Tanaka, "Universal designated-verifier signature with aggregation," in *Proc. Third International Conf. Inf. Technol. Appl.*, 2005.
[22] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology: Proc. Crypto'84*, Lecture Notes in Computer Science vol. 196, (Berlin), pp. 47-53, Springer-Verlag, 1985.
[23] W. Diffle and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, pp. 644-654, 1976.
[24] E. Biham, D. Boneh, and O. Reingold, "Breaking generalized Diffie-Hellman modulo a composite is no easier than factoring," *Inf. Process. Lett.*, vol. 70, pp. 83-87, 1999.
[25] T. A. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469-472, 1985.
[26] L. Harn and Y. Xu, "Design of generalized ElGamal type digital signature schemes based on discret logarithm," *Electron. Lett.*, vol. 30, no. 24, pp. 2025-2026, 1994.
[27] R. Lidl and H. Niederreiter, *Finite Fields*. Cambridge University Press, 2000.
[28] L. Harn, J. Ren, and C. Lin, "Design of DL-based certificateless digital signatures," *J. Syst. Software*, vol. 82, pp. 789-793, 2009.
[29] A. Shamir and Y. Tauman, "Improved online/offline signature schemes," in *Proc. 21st Annual International Cryptology Conf. Advance Cryptology*, p. 355-367, Springer-Verlag, 2001.
[30] H. Krawczyk and T. Rabin, "Chameleon signatures," in *Proc. Symp. Netw. Distributed Syst. Security (NDSS00), (Internet Society)*, pp. 143-154, Feb. 2000.

Dr. Lein Harn received his Bachelor of Science degree in Electrical Engineering from the National Taiwan University in 1977. In 1980, he received his MS in Electrical Engineering from the State University of New York-Stony Brook and in 1984 he received his doctorate degree in Electrical Engineering from the University of Minnesota. He joined as an Assistant Professor in the department of Electrical and Computer Engineering at the University of Missouri-Columbia in 1984 and in 1986, he moved to Computer Science and Telecommunication Program (CSTP) of University of Missouri-Kansas City (UMKC). While at UMKC he went on development leave to work in Racal Data Group in Florida for a year. His research interests are cryptography, network security and wireless communication security. He has published number of papers on digital signature design and applications, wireless and network security. He has written two books on Security. At present he is investigating new ways of using digital signature in various applications.



Jian Ren received the B.S. and M.S. degrees both in mathematics from Shaanxi Normal University, China, in 1988 and 1991 respectively. He received the Ph.D. degree from Xidian University in 1994. From 1997 to 1998, he was with Racal Datacom as a security architect. From 1998 to 2002, he was first with Bell-Labs and later with Avaya Labs as a member of technical staff. He joined the Department of Electrical and Computer Engineering department at Michigan State University in 2005 as an assistant professor. His current research interests are in the areas cryptography, network security, energy efficient sensor network security protocol design, and privacy-preserving communications. He is a recipient of the U.S. National Science Foundation Faculty Early Career Development (CAREER) award in 2009. He is a senior member of IEEE.