Interactive Zero Knowledge Password Authentication Scheme for Commercial Web Sites

Satish M. Srinivasan
School of Graduate Professional Studies
Penn State University
Malvern, PA 19355

Indranil Sengupta
Indian Institute of Technology,
Kharagpur, WB 721302

Pratap K. J. Mohapatra
Indian Institute of Technology,
Kharagpur, WB 721302

ABSTRACT
This paper presents the implementation of an interactive Zero Knowledge Password authentication scheme for commercial Web sites. In this scheme, a legitimate prover (client) can exchange a secret code (password) with a remote skeptic (server), in order to reveal his/her identification. Based on the validity of the secret code the skeptic then allows the prover to login to the site and access the web services. This paper introduces a protocol that integrates the concepts of Discrete Logarithm Problem (DLP) and Zero-Knowledge Proofs (ZKP). The protocol consists of three entities, namely, the prover, the skeptic, and the facilitator who interact with one another to generate the secret code. When tested, the time to carry out various operations related to this protocol was reasonably small (under 4 seconds). Our scheme is resistant to man-in-the-middle attack and discourages replaying previously intercepted secret codes. We also propose two modifications to our basic scheme to make it resistant against the attack on Integrity and Denial of Service attack (DOS).

General Terms
Discrete Logarithm Problem (DLP), Zero-Knowledge Proofs (ZKP), Dynamic on-Demand Password (DDP).

Keywords
Discrete Logarithm Problem (DLP), Zero-Knowledge Proofs (ZKP), Dynamic on-Demand Password (DDP), Authentication, commercial web sites and Denial of Service (DOS) attack

1. INTRODUCTION
Modern developments in computer and communication technologies has provided people with an option to communicate over networks and run their jobs on remote hosts. Although convenient procedures for communication exist, due to their distributed nature, the networks fall prey in the hands of eavesdroppers. Privacy and security has thus become increasingly important [10]. Security is an important attribute for most commercial sites, especially for those who deal with selling and buying of commodities, auctioning of goods, and with electronic payments. There are many aspects of security: authentication, confidentiality, distribution of secret information, data integrity, availability, and non-repudiation [4]. This paper addresses only the authentication aspect. Usually, for the purpose of authentication, securing the login and password information of a prover is confined to the use of encryption and decryption facility with the help of a secret key or a public key system. These systems are safe until the secret information is sabotaged or is not intelligible by some malicious personality.

Passwords are still the most common basis for user authentication. Even systems with sophisticated cryptographic protocols often employ user passwords. It seems likely that passwords will continue to be in use for quite some time [7]. Traditionally, password authentication schemes require that whenever a user logs in, the submitted password is verified with the already stored password table in the system. Although this method can prevent the passwords from being disclosed, there are several issues. This method is not resistant to replaying previously intercepted password and the contents of the plain password table can be modified by a malicious personality [10]. According to Taekyoung [7], the user passwords have very low entropy and they are hard to transmit securely over an insecure channel. Secondly, the password files are hard to protect. Taekyoung further suggests that, to be effective, this password-based solution should have an amplified password scheme and an amplitude password file, which is similar to the concepts underlining the Zero-Knowledge proofs. Taekyoung describes a new efficient password-based protocol for defeating the guess-based attack. Their protocol uses a one-time pad to encrypt the session key securely and a strong one-way hash function for integrity.

Yang and Shieh [8] have proposed a time-stamp based password authentication scheme. In this scheme, the users are allowed to choose and change their passwords freely, and the remote host is not required to maintain a password table or a verification table for verifying the legitimacy of the login users. Molva and Tsudik [9] state that the traditional methods of user authentication suffer from an important weakness arising out of the low degree of randomness in secret code that human beings can use for identification. These secret codes, though not exposed over the communication lines, are vulnerable to off-line brute force attacks based on exhaustive trials. Molva and Tsudik also say that these secret codes are chosen from a relatively small key space and that any determined adversary can break the password by a trial-and-error method to find a match between the trial value and the message.

Gritzalis and Katsikas [2] state that the key in controlling access to a computerized information system is to establish a positive and a unique identification for every user to whom the access is to be granted. They criticize the communication channels for being the weakest links in the password mechanism and emphasize the need for cryptographic techniques such as Zero Knowledge Proofs and Probabilistic Login procedures to make the message unintelligible and allow the skeptic to gain enough information about the identity of the prover without revealing the knowledge owned by the prover. They recommend quadratic residue problem for implementing Zero Knowledge Proofs and have implemented the scheme for system-to-system authentication. However, it allows limited user participation due to the relatively large computation time requirement.

In this paper, we have presented a scheme for login authentication that uses Discrete Logarithm Problem (DLP), which has been

solved by using the concepts of Zero Knowledge Proofs [12][13] (referred by [3]). We have also discussed the implementation features of the scheme in JAVA.

2. DISCRETE LOGARITHM PROBLEM (DLP) AND ZERO-KNOWLEDGE PROOFS (ZKP)

We have chosen the Discrete Logarithm Problem (DLP) because it is a NP-Complete problem [3][5]. The skeptic can easily verify a solution to a DLP given by a prover; however, the skeptic, who does not know the solution to the DLP problem, will not be able to solve the problem in polynomial time.

In simple language, a DLP is to find an integer x such that $\alpha^x \equiv \beta \pmod{C}$, where **C** is prime and x is relatively prime to **C**. Koblitz [5] gives the following more rigorous definitions: "Let G be a finite cyclic group of order n. Let α be a generator of G, and let $\beta \varepsilon G$. The discrete logarithm of β to the base α, denoted by $\log_\alpha \beta$, is the unique integer x, $0 <= x <= n-1$, such that $\beta = \alpha^x$". The discrete logarithm problem (DLP) is the following: given a prime C, a generator α of Z_C^*, and an element $\beta \varepsilon Z_C^*$, find an integer x, $0 <= x <= C-2$, such that $\alpha^x \equiv \beta \pmod{C}$.

Zero-knowledge proofs (ZKP) are formal methods that give the prover the ability to validate his/her identity using a secret password, without actually revealing it. The DLP, which is an NP-Complete problem, has a Zero Knowledge Proof [3]. Schneier [3] suggests an interactive method where the prover gives values of all the parameters appearing in the **DLP** except for the unknown solution x. Thereafter a sequence of to-and-fro communications takes place between the skeptic and the prover. The skeptic sends a set of random numbers to the prover who generates a number Z. Z is a function of x that correctly solves another **DLP,** similar to the original **DLP**, to the satisfaction of the skeptic.

3. THE AUTHENTICATION SCHEME

The authentication scheme presented here has three phases: the Certificate initialization phase, the Login phase and the Verification phase. In the Certificate verification phase the **prover (P)** sends his/her knowledge to the **skeptic (S)** and the skeptic issues a login certificate to the prover. In the Login phase, upon receiving a request from the prover for remote login, the skeptic sends a set of questions based on the knowledge submitted by the prover in the previous phase. The prover, aided by a **facilitator (F)**, determines the solution for the questions and returns the solution set to the skeptic. A facilitator is a tool that performs highly time-consuming computations, thus relieving the prover from performing complex computations. The latter submits a question string and the Login certificate to the facilitator requesting for a solution set. Considering the changing nature of the question-solution set and the password-like function it serves, we have named the solution set generated by the prover with the aid of the facilitator a **Dynamic on-Demand Password (DDP).** After receiving the solution set from the prover in the Verification phase, the skeptic verifies the solution and, based upon its judgment on the authentic/malicious nature of the prover, returns him/her the login status (Yes/No). The system sequence diagram in Figure 1 depicts the interactions between the prover, the facilitator, and the skeptic in all the three phases. The details for each of the three phases are:

Certificate Initialization Phase – The prover, with the help of the facilitator, interacts with the skeptic and establishes a Login Certificate by following the steps of the RSA-based Massey-Omura Cryptosystem scheme given in Table 1 [5]. In this scheme, the various keys (512 bits) used for encryption and decryption by the prover and the skeptic are: $\{\mathbf{KU_p}, \mathbf{KR_p}\}$ and $\{\mathbf{KU_s}, \mathbf{KR_s}\}$. $\mathbf{KU_p}$ is the public key of the prover and $\mathbf{KU_s}$ is the public key of the skeptic. The corresponding private keys of the prover and the skeptic are $\mathbf{KR_p}$ and $\mathbf{KR_s}$ respectively. It is assumed here that the prover and the skeptic have already established a key pair from the Trusted Third Party (TTP). $\{\mathbf{EU_p}, \mathbf{DR_p}\}$ and $\{\mathbf{EU_s}, \mathbf{DR_s}\}$ are the generated encryption and decryption keys of the prover and the skeptic respectively. The generated encryption and decryption keys are destroyed after the Massey-Omura Cryptosystem scheme is completed successfully. The encryption and decryption are performed using the **RSA** algorithm [1][3][5][6]. The prover decides on the values of the parameters of the **DLP** ($\alpha^x \equiv \beta \pmod{C}$, where Z^*_C *is of order* γ and sends them to the skeptic, keeping the **x** secret. In addition, the prover also sends his user name hashed with a hashing function **H,** (i.e. MD5 [3]) and a random string **R**. The set of **DLP** parameters and the hashed user name is collectively termed as **DATA** (see figure 1). After performing several computations by the prover and the skeptic, and establishing several to-and-from communications between them (see table 1), the skeptic establishes a login certificate for the prover.

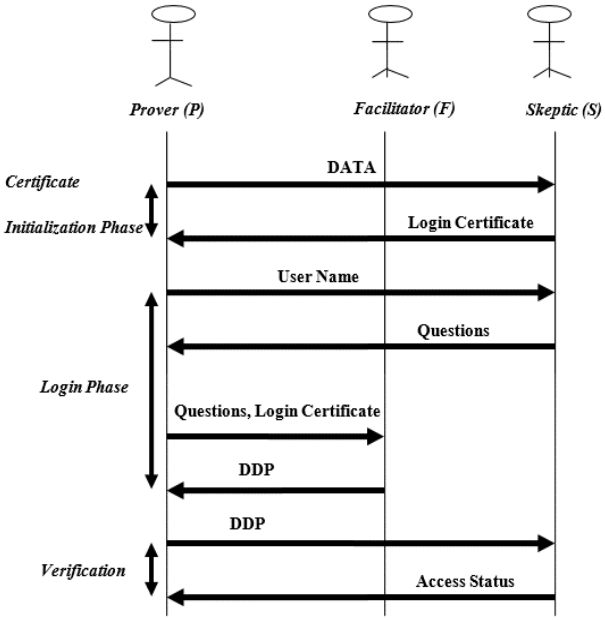

Fig 1: System Sequence Diagram for the Selected Authentication Scheme

Table 1. Message Computation and Transmission in the Massey-Omura Cryptosystem Scheme

Computation	Transmission
P: E_{EUp}[DATA], E_{KUs}[R]	**P** → **S**: {E_{EUp}[DATA], E_{KUs}[R] }
S: $D_{KRs}[E_{KUs}[R]] = [R']$ $E_{KUp}[R']$, $E_{EUs}[E_{EUp}[DATA]]$	**S** → **P**: {$E_{KUp}[R']$, $_{EUs}[E_{EUp}[DATA]]$}
P: $D_{KRp}[E_{KUp}[R'']] = [R'']$ if $[R'']=[R']$ then $D_{DRp}[E_{EUs}[E_{EUp}[DATA]]]$	**P** → **S**: {$D_{DRp}[E_{EUs}[E_{EUp}[DATA]]]$}
S: $D_{DRs}[D_{DRp}[E_{EUs}[E_{EUp}[DATA]]]]$ = DATA	

In Table 1, **R** is a random string generated by the prover, **R'** is the random string received by the skeptic from the prover, and **R''** is the random string received by the prover from the skeptic, **DATA** = {**H (User Name), α, β, γ, C**}, where **H** is a one-way function. $\mathbf{E_{KUp}}$**(DATA**) is the Login Certificate, where $\mathbf{E_{KUp}}$ is the

public key of the prover.

Login Phase

DDP Generation - The login phase starts when the prover makes a request for login by giving his/her user-id. The skeptic thereafter generates a question string of random bits $\mathbf{Q = \{0, 1\}^+}$ of length **n** by using the Blum Blum Shub generator (BBS) [1], and sends it to the prover. The prover shares **Q** and the login certificate with the facilitator to form a password called **DDP**. The **DDP** is a set of (**2 * n + 1**) elements. Table 2 demonstrates the steps in the Login phase. We have followed the Zero-Knowledge proof of a Discrete Logarithm Problem, suggested by [12][13], (referred by [3]), to generate its solution.

Table 2: Steps in the Log-in Phase: DDP Generation

Input:
(**A (α), B (β), C**) where $A^x \equiv B \ (mod\ C)$, $\mathbf{Q = \{0, 1\}^+}$.

Computation:
P: Receives the question string $\mathbf{Q = \{0, 1\}^+}$ of length **n** from the skeptic**.**
P: Generates **n** random numbers $\mathbf{r_1, r_2, ---, r_n}$ where all $\mathbf{r_i < C-1}$.
P: Computes and sends $\mathbf{h_i = A^{r}_i \bmod C}$, for all **i** = **1** to **n**.
P: Replies to all questions $\mathbf{Q_i}$, **i =1, …, n**.

- If $\mathbf{Q_i = 0}$, then $\mathbf{r_i}$.
- If $\mathbf{Q_i = 1}$, then $\mathbf{s_i = (r_i - r_j) \bmod (C - 1)}$, **j** is the lowest value of the random number $\mathbf{r_i}$ for which $\mathbf{Q_i = 1}$.

P: Sends $Z = (X - r_j) \bmod (C-1)$.

Output:
DDP: $\mathbf{\{h_{1...}\ h_n, \{s_i \text{ or } r_i\}\ i=1 \text{ to } n, Z, E_{KRp}[h\ \{h_{1...}\ h_n, \{s_i \text{ or } r_i\}\ i= 1 \text{ to } n, Z\}]\}}$ **where** $\mathbf{h_{1...}\ h_n, \{s_i \text{ or } r_i\}\ i=1 \text{ to } n, Z < C.}$ $\mathbf{KR_p}$ is the private key of the prover.

Facilitator - In the Certificate Initialization phase, the facilitator helps the prover in establishing the parameters for the DLP problem (***α, β, γ, C***}. During the Login phase, the facilitator verifies the login certificate submitted by the prover, and generates a **DDP** based upon the knowledge available in the login certificate and the questions provided by the prover.

Verification Phase - The skeptic verifies the validity of the DDP submitted by the prover and communicates the access status of the prover for the particular login request. Table 3.1 demonstrates the steps in the Verification phase. Various operations performed in these three phases are Hashing, RSA Encryption, RSA Decryption, and RSA Key Generation. Table 3.2 gives the operations performed by prover and skeptic and the frequencies of these operations in all the three phases.

4. IMPLEMENTATION OF OUR SCHEME

The scheme described in the previous section was implemented in JAVA. The prover and the skeptic programs were executed in the same machine to avoid any communication delays. The scheme particularly targeted the login authentication of the client. The following steps detail the working of the authentication scheme in various phases (See Table 4):

Table 3.1: Verification Phase

Input:
DDP: $\mathbf{\{h_{1...}\ h_n, \{s_i \text{ or } r_i\}}_{\ i=1 \text{ to } n}$**, Z,** $\mathbf{E_{KRp}[h\ \{h_{1...}\ h_n, \{s_i \text{ or } r_i\}}_{\ i=1 \text{to } n}$**, Z}]} where** $\mathbf{h_{1...}\ h_n, \{s_i \text{ or } r_i\}\ i=1 \text{ to } n, Z < C.}$ Where $\mathbf{KR_p}$ is the private key of the prover.

Computation:
S: Verifies the following equivalence for every question $\mathbf{Q_i}$, **i= 1, …, n** where $\mathbf{Q_i = \{0, 1\}^+}$:
If $\mathbf{Q_i=0}$, then $\mathbf{A^{ri} \equiv h_i \ (mod\ C)}$.
If $\mathbf{Q_i=1}$, then $\mathbf{A^{si} \equiv h_i h_j^{-1} \ (mod\ C)}$.
S: Verifies if $\mathbf{A^Z \equiv B h_j^{-1} \ (mod\ C)}$.

Output:
Access status (Y/N)

Table 3.2: Frequencies of various operations performed by the prover (P) and the skeptic (S)

Types of operation	Certificate initialization phase	Login phase	Verification phase
Hashing	P: 1 S: 0	P: 1	S: 1
RSA Encryption	P: 2 S: 2	P: 1	S: 0
RSA Decryption	P: 2 S: 2	P: 0	S: 1
RSA Key Generation	P: 1 S: 1	P: 0	S: 0

Table 4: Protocol implementation

Certificate Initialization Phase

Prover:
Generates the parameter **A, B, γ, C** for a known **x** in $\mathbf{A^x \equiv B \ (mod\ C)}$, where Z^*_C is of order **γ. [**See definition of **DLP** in section 2**]**

Skeptic:
Issues a Login Certificate. **[**See Table 1**]**

Login Phase

Prover:
Sends the user name.

Skeptic:
Generates a string of random bits of length **n** using the BBS generator [1] and sends it to the prover.

Prover:
The Facilitator aids the prover in executing the following steps:

- Generate **n** random numbers $\mathbf{r_1, …, r_n}$.
- Computes, for **i= 1 to n**, $\mathbf{h_i = A^{ri} \bmod C.}$
- Computes $\mathbf{r_i}$ or $\mathbf{s_i}$ corresponding to every $\mathbf{i^{th}}$ element of the question string **Q**, $\mathbf{r_i}$ if **0** and $\mathbf{s_i}$ otherwise. [See Table 2]
- Computes **Z** as a function of **x**. [See Table 2]
- Compiles a solution set of length (**2*n + 1**), where the first **n** elements are $\mathbf{h_i}$**'**s, the next **n** elements are either $\mathbf{r_i}$**'**s or $\mathbf{s_i}$**'**s, and the last element is **Z**.
- Hashes all (**2 * n + 1**) elements of the solution set, encrypts the hashed output with the prover's private key and adds encrypted output as an addendum to the compiled solution set of length (**2*n + 1**), to form a **DDP**.

Prover:
The prover sends the **DDP** to the skeptic**.**

Verification Phase

Skeptic:
The skeptic executes the following steps:

- Decrypts the encrypted message using the prover's public key.
- Creates a hash of the first (**2 * n + 1**) elements sent by the prover.
- Verifies if the hashed output matches the decrypted message.

If the hashed output and the decrypted message is same

- Verifies that $\mathbf{r_i}$'s and $\mathbf{s_i}$'s are solutions to the simple **DLP** problems. [See Table 3.1]
- Verifies that **Z** is the solution to a **DLP** problem. [See Table 3.1]
- Sends the access status (Yes/No) to the prover.

5. TESTING OF THE PROTOCOL

For testing, we assume that both the prover and the skeptic are aware of the login certificate. The Login Certificate used is {36015f02e80ce8b88d9a8b0f45fcc903, 2, 228, 191, 383)}, where the first element is the hash of a user name generated using a hash function, and the remaining elements are the parameters **α, β, γ, C.** The skeptic sends to the prover a question string (**Q**) of bit length 5: {0, 1, 0, 1, 1}. The prover provides the login certificate (**DATA**) and the question string (**Q**) to the facilitator. The facilitator, in turn, generates a solution set {0, 11, 186, 99, 150, 67, 138, 40, 16, 128, 63} as described in the Log-in Phase. The facilitator then computes the **DDP**, which is:

{0, 11, 186, 99, 150, 67, 138, 40, 16, 128, 63,
841791982449889344254219564237643845940003586155916500423636204066028313050584498419554739356780322498488608613500325096203476840006631402868760599 9}.

Here the first five elements correspond to $\mathbf{h_i}$**'s,** the next five elements to either $\mathbf{r_i}$**'s** or $\mathbf{s_i}$**'s**, and the eleventh element to **Z**. The twelfth element is the encryption of the hash of the solution set represented by the eleven elements, and is generated by using the private key of the prover. The prover then sends the **DDP** to the skeptic. Upon receiving the **DDP**, the skeptic performs the required steps in the verification phase.

Table 5.1 gives an account of the time taken to generate the **DDP** and Table 5.2 gives an account of the time taken to verify the **DDP**. The total time taken to generate **DDP** was 3.645 seconds, and for verification of **DDP**, it took only 0.091 second. This scheme was implemented and tested on a 64-bit windows machine with Intel core i5 processor @2.40 GHZ with 16 GB RAM.

Table 5.1: Timings for various operations in the Login Phase.

Operation Type	Time Taken (Sec)
Generating the hash code	0.02
Decrypting the Login certificate (RSA decryption using Chinese remainder theorem) [5]	0.00016
Matching Private key	0.0006
Generating the 17 parameters + Hash code generation + encryption of the hash code	3.615 + 0.020 + 0.010 = 3.645
Total time for password generation (Login)	3.64576

Table 5.2: Timings for various operations in the Verification Phase.

Operation Type	Time Taken (Sec)
Generating the hash code	0.010
Decrypting the second part of the DDP	0.030
Verifying the hash codes	0.001
Verifying the first part of DDP.	0.050
Total time taken for verifying DDP (Authentication)	0.091

6. CONCLUSION

We have proposed an interactive zero-knowledge password authentication scheme that differs from other traditional password authentication scheme in that the password involves a long set of numerical characters, is short lived, and is based on knowledge. The scheme provides two major advantages: one, there is no need to maintain any password table at the skeptic end, and two, the short-lived nature and the uninformative nature of the password discourages replay attacks. The result shows that the password generation time is relatively less and the password verification time is negligibly small.

It is not easy to prove formally that a cryptographic algorithm is secure [11]. Taekyoung [7], however, has stated the superiority of the zero-knowledge proof based authentication scheme to be more resistant to malicious attacks, compared to the traditional schemes. The proposed scheme is resistant to both the man-in-the-middle attack and the reply attack. In the former, after accessing the DDP sent by the prover in the Login Phase, the saboteur can decrypt it using the prover's public key. However, to maintain the structure of the DDP, he/she has to re-encrypt the DDP with the public key of the prover (not the private key of the prover which is required to generate the DDP), thus making the DDP intractable by the skeptic. The compact nature and the intricate complexity of the **DDP** structure make it resistant to the man-in-the-middle attack. As for as the replay attacks are concerned, the questions posed by the skeptic are random and differ for each login attempts, thus changing the contents of the DDP, and making the scheme inherently resistant to replay attacks.

The scheme can also be modified to counter the Denial of Service and integrity attacks. In the former, an encrypted version of the digital signatures can be introduced in the DDP structure. Thus, following the logic for the man-in-the-middle attack, the saboteur will fail to substitute a correct DDP structure, making the DDP intractable to the skeptic. As far as the attacks on integrity is concerned, the skeptic provides the prover, along with the question string, a string representing the date and time of the particular login request. The prover adds the following string in the DDP, thus changing the contents of the DDP, and sends it to the skeptic.

As we have seen, the Zero-Knowledge based Dynamic on-Demand Password provides great resistance to various attacks while allowing access to the genuine user and takes reasonably less time. The proposed scheme thus has great potential for being adopted for authentication in commercial Web sites. We are in the process of implementing this scheme for web sites that will support online auction of antique items. This work shall be reported on a later date.

7. REFERENCES

[1] W. Stallings, Cryptography and Network Security - Principles and Practice, Second ed., Pearson Education Asia, Prentice Hall, NJ, 1995.

[2] D. Gritzalis, S. Katsikas, Towards a formal system-to-system authentication protocol, Computer Communication, 19 (1996) 954-961.

[3] B. Schneier, Applied Cryptography, Second ed., John Wiley & Sons, Inc., New York, 1996.

[4] D.A. Menasce, A.F. Almeida, Scaling for E-Business, First ed., Prentice Hall, NJ, 1998.

[5] N. Koblitz, A Course in Number Theory and Cryptography, Second ed., Springer–Verlag, New York, 1994.

[6] A. Menezes, P.V. Oorschot, S. Vanstone, Handbook of Applied Cryptography, Fifth ed., CRC Press, 2001.

[7] K. Taekyoung, Authentication and key agreement via memorable password, Computer Communication, 11 (1989) 753-771.

[8] W.H. Yang, S.P. Shieh, Password authentication scheme with smart cards, Computer and Security, 18 (1999) 727-733.

[9] R. Molva, G. Tsudik, Increased randomness in modern password scheme, Computer Communication, 31 (1998) 753-762.

[10] K. Tan, H. Zhu, Remote password authentication scheme based on cross-product, Computer Communication, 22 (1999) 390-393.

[11] T.C. Wu, Remote login authentication scheme based on a geometric approach, Computer Communication, 18 (1995) 959-963.

[12] D. Chaum, J.-H. Evertse, J. van de Graff, An improved protocol for demonstrating possession of discrete logarithms and some generalizations, Advances in Cryptology – Eurocrypt ’87 Proceedings, Springer-Verlag, New York, (1988) 127-141.

[13] D. Chaum, J.-H. Evertse, J. van de Graff, Demonstrating possession of discrete logarithm without revealing it, Advances in Cryptology- Eurocrypt ’86 Proceedings, Springer-Verlag, New York, (1987) 200-212.