INFORMATICA, 2004, Vol. 15, No. 2, 161–170
© 2004 *Institute of Mathematics and Informatics, Vilnius*

Improvement on Pretty-Simple Password Authenticated Key-Exchange Protocol for Wireless Networks

Ting-Yi CHANG
Department of Computer and Information Science
National Chiao-Tung University
1001 Ta Hsueh Road, Hsinchu, Taiwan, R.O.C.
e-mail: tychang@cis.nctu.edu.tw

Chou-Chen YANG
Department and Graduate Institute of Computer Science and Information Engineering
Chaoyang University of Technology
168, Gifeng E. Rd., Wufeng, Taichung County, Taiwan 413, R.O.C.
e-mail: ccyang@mail.cyut.edu.tw

Chia-Meng CHEN
Graduate Institute of Networking and Communication Engineering
Chaoyang University of Technology
168, Gifeng E. Rd., Wufeng, Taichung County, Taiwan 413, R.O.C.
e-mail: s9130601@mail.cyut.edu.tw

Received: September 2003

Abstract. This paper presents an improved method inspired by the recently proposed Pretty-Simple PAKE (Password Authenticated Key-Exchange) protocol, which is already a well-known, robust and simple password authenticated key exchange scheme. In our even more efficient scheme, only the password needs to be negotiated in advance, and the computations have also been simplified. Our scheme is based on the elliptic curve discrete logarithm problem and can gain the benefit from the key block size, speed and security. Since our new scheme is both efficient and fast with a low cost for device storage, it is especially suitable for the wireless network environment.

Key words: elliptic curve discrete logarithm, password authentication, key exchange, wireless network.

1. Introduction

Nowadays, wireless network technologies have seen quite a few innovations and gained amazing popularity. In the wireless environment, applications and users can enjoy higher mobility levels and greater convenience than in the wired environment. However, on the other hands, the lacking of security shelter and sufficient bandwidth raises secure problems in data transmission.

The cryptosystem has recently been a focus of importance in the field of wireless data transmission. Symmetric cryptosystems such as DES and asymmetric cryptosystems such as RSA (Rivest, 1978) and ELGamal (ElGamal, 1985) are the most popular methods in data encryption. But the above-mentioned methods need to negotiate mutual keys before they can encrypt/decrypt valuable information when initiating a conversation between two unconfirmed parties.

In the well-known Diffie–Hellman key change algorithm (Diffie and Hellman, 1976), since both sides intend to securely communicate with each other, they exchange some exposed information to generate a mutual key (session key) that encrypts the valuable messages only comprehended by both parties.

Diffie–Hellman's method can easily produce a session key, but it suffers from the crash by the man-in-the-middle attack. In order to prevent such an attack, several password authenticated key exchange (PAKE) schemes have been presented in (Goldreich and Lindell, 2001; Katz *et al.*, 2001; Kobara and Imai, 2002). The Pretty-Simple PAKE protocol (Kobara and Imai, 2002) possesses satisfactory characteristics such as robustness and simplicity in their three phases. In that scheme, two entities use the password to produce the keying material in the secrecy-amplification phase, and then both sides will create the session key in the session-key generation phase after they complete the verification phase.

Unfortunately, the Pretty-Simple PAKE protocol faces some problems when it is applied to the wireless environment. The protocol brings a heavy load to the computation and requires both sides to store several extra parameters. These shortcomings restricts the performance of the protocol in the wireless environment because restricted bandwidth and small storage sizes are exactly the bottlenecks of wireless systems.

In recent years, the literature has shown that the Elliptic Curve Cryptosystem (ECC) has become an attractive alternative cryptosystem because its security is based on the elliptic curve discrete logarithm problem (ECDLP) (Koblitz, 1987; Miller, 1986). ECC operates over a group of points on an elliptic curve and offers a level of security comparable to classical cryptosystems that use much larger key sizes (Koblitz *et al.*, 2000; Menezes, 1993; Menezes and Vanstone, 1995; Miller, 1986; Schroeppel *et al.*, 1995).

In this paper, we shall propose a much simpler method based on the ECDLP that has gained benefits from the key block size, speed and security. Furthermore, in our new method, only the password needs to be negotiated in advance. Since our protocol has both strengths in efficiency and cost of memory space, it is suitable for the wireless network environment.

The rest of this paper is organized as follows. The next section reviews the Pretty-Simple PAKE protocol. Then, in Section 3, we shall describe our improved method in detail, followed by the security analysis in Section 4. The comparison between our scheme and the Pretty-Simple PAKE protocol will be discussed in Section 5, and the conclusion of this paper will be made in the last section.

2. Review of Pretty-Simple PAKE Protocol

In this section, we shall briefly review the Pretty-Simple PAKE protocol and discuss the defects which occur in the wireless network environment. In the registration stage, the client first acquires a mutual password and three distinct values (Tag_s, Tag_c, Tag_{sk}) after it registers with the server. This can be accomplished through a secure channel. The protocol is defined over a finite cyclic group $G = \{g^i \bmod p\colon\ 0 <= i <= q\}$, where p is a large prime number, and q is a large prime divisor of $p - 1$. Both g_1 and g_2 are public generators chosen in the registered stage. The session key generation process is depicted in Fig. 1, and the detailed steps are described as follows:

Step 1. $Client \longrightarrow Server$: $Y_c = g_1^{r_c} \cdot g_2^{pass}$.

The client first chooses a random number $r_c \in (Z/qZ)^*$ to calculate $Y_c = g_1^{r_c} \cdot g_2^{pass}$, $pass$ is a mutual password. Then the client sends Y_c to the server directly.

Step 2. $Server \longrightarrow Client$: $Y_s = g_1^{r_s} \cdot g_2^{pass}$.

Using its mutual password $pass$ and a random number $r_s \in (Z/qZ)^*$, the server also calculates $Y_s = g_1^{r_s} \cdot g_2^{pass}$. Then the server sends Y_s to the client directly.

Step 3. $Client \longrightarrow Server$: $V_c = MAC_{K_c}(Tag_c \parallel Y_c \parallel Y_s)$.

The client can gain the keying material $K_c = (Y_s \cdot g_2^{-pass})^{r_c}$ after receiving Y_s. The client uses a MAC (Message Authentication Codes) generation function (Naor and Yung, 1989) and keying materials as its key. A pre-determined distinct value $Tag_c = (ID_c \parallel ID_s \parallel 00)$, in which ID_c and ID_s are the identities of the client and the server, and $\parallel$ denotes the concantation. The client uses Y_c and Y_s together with Tag_c to calculate $V_c = MAC_{K_c}(Tag_c \parallel Y_c \parallel Y_s)$, and then it sends V_c to the server.

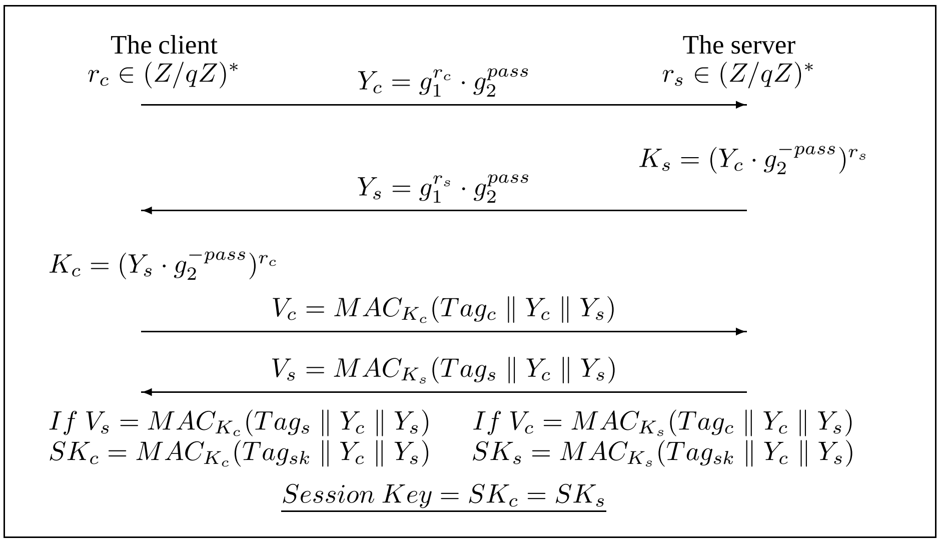
The client
$r_c \in (Z/qZ)^*$
$Y_c = g_1^{r_c} \cdot g_2^{pass}$
The server
$r_s \in (Z/qZ)^*$
$K_s = (Y_c \cdot g_2^{-pass})^{r_s}$
$Y_s = g_1^{r_s} \cdot g_2^{pass}$
$K_c = (Y_s \cdot g_2^{-pass})^{r_c}$
$V_c = MAC_{K_c}(Tag_c \parallel Y_c \parallel Y_s)$
$V_s = MAC_{K_s}(Tag_s \parallel Y_c \parallel Y_s)$
$If\ V_s = MAC_{K_c}(Tag_s \parallel Y_c \parallel Y_s)$
$SK_c = MAC_{K_c}(Tag_{sk} \parallel Y_c \parallel Y_s)$
$If\ V_c = MAC_{K_s}(Tag_c \parallel Y_c \parallel Y_s)$
$SK_s = MAC_{K_s}(Tag_{sk} \parallel Y_c \parallel Y_s)$
$\underline{Session\ Key = SK_c = SK_s}$

Fig. 1. Pretty-Simple PAKE protocol.

Step 4. $Server \longrightarrow Client$: $V_s = MAC_{K_s}(Tag_s \parallel Y_c \parallel Y_s)$.

The server can gain the keying material $K_s = (Y_c \cdot g_2^{-pass})^{r_s}$ after receiving Y_c. As in Step 3, the server uses Y_s, Y_c and $Tag_s = (ID_c \parallel ID_s \parallel 01)$ to calculate $V_s = MAC_{K_s}(Tag_s \parallel Y_c \parallel Y_s)$, and then the server sends V_s to the client.

After both parties exchange V_c and V_s, they can verify $V_s = MAC_{K_c}(Tag_s \parallel Y_c \parallel Y_s)$ and $V_c = MAC_{K_s}(Tag_c \parallel Y_c \parallel Y_s)$. If the verification is successful, they generate their session keys $SK_c = MAC_{K_c}(Tag_{sk} \parallel Y_c \parallel Y_s)$ and $SK_s = MAC_{K_s}(Tag_{sk} \parallel Y_c \parallel Y_s)$, respectively.

Based on the discussions in (Kobara and Imai, 2002), we can say that the Pretty-Simple PAKE protocol has several advantages over most other PAKE methods at present. The protocol outperforms others in both robustness and simplicity. However, it still has to face some problems when applied to wireless networks because of the low power and computational capability of wireless devices. The discussions are as follows:

- The above-mentioned protocol depends on the difficulty of solving DLP (Discrete Logarithm Problem) (Diffie and Hellman, 1976). On the other hand, it also exponentially increases the load on calculation. Obviously since it deploys two generators g_1 and g_2, the computation loading increases significantly.
- In the above-mentioned protocol, two parties must negotiate several parameters in addition to the password in advance. Each client must record Tag_c, Tag_s and Tag_{sk} additionally and put them to use during the verification and session-key generation phase. This increases the burden on the system capacity.
- Too many steps weaken the performance of the devices and network. Fewer steps can conserve more bandwidth in the wireless environment, and therefore the time needed for the session key generation stage can be shortened as well.

According to the discussions above, the Pretty-Simple PAKE protocol does not seem to be an appropriate solution to the wireless environment. In the next section, we will show an improved scheme which retains the concept of keying material and refines the composition of the Pretty-Simple PAKE protocol so that the refined protocol will be suitable for the wireless environment.

3. Our Proposed Scheme

The intention of this paper is to suggest an appropriate scheme for the wireless network environment. When the client wants to initiate a secure conversation with the server or exchange some valuable information with no one else except the server, they encrypt the plaintext by using the mutual session key which is generated by both sides. We borrow the concept of keying material in the Pretty-Simple PAKE protocol, and therefore the password can be protected in this stage. After the client registers with the server, the server chooses an elliptic curve E over a finite field $GF(q)$. The client and the server only need to store a mutual password. The session key generation steps are described as follows, and a sketchy depiction is in Fig. 2:

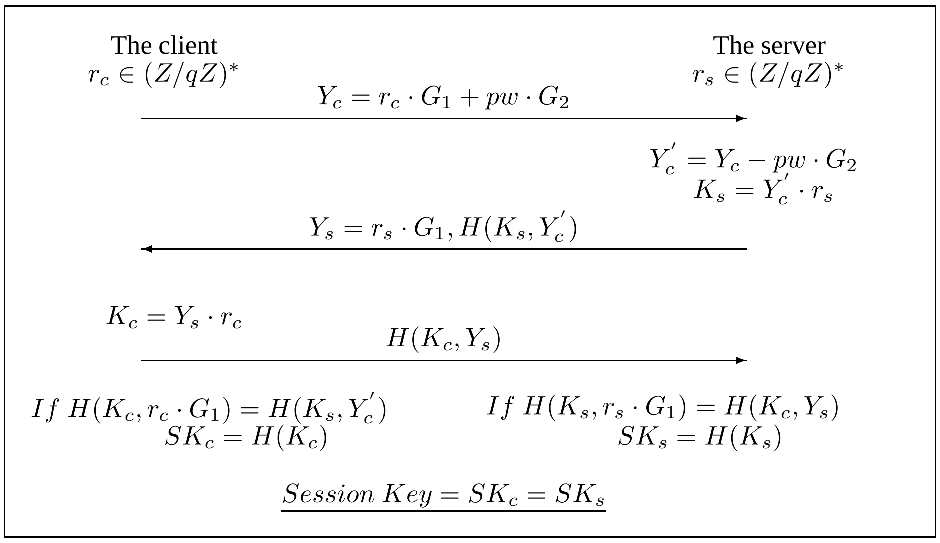
The client
$r_c \in (Z/qZ)^*$
The server
$r_s \in (Z/qZ)^*$
$Y_c = r_c \cdot G_1 + pw \cdot G_2$
$Y_c^{'} = Y_c - pw \cdot G_2$
$K_s = Y_c^{'} \cdot r_s$
$Y_s = r_s \cdot G_1, H(K_s, Y_c^{'})$
$K_c = Y_s \cdot r_c$
$H(K_c, Y_s)$
$If\ H(K_c, r_c \cdot G_1) = H(K_s, Y_c^{'})$
$SK_c = H(K_c)$
$If\ H(K_s, r_s \cdot G_1) = H(K_c, Y_s)$
$SK_s = H(K_s)$
$\underline{Session\ Key = SK_c = SK_s}$

Fig. 2. Our proposed scheme.

Step 1. $Client \longrightarrow Server$: $Y_c = r_c \cdot G_1 + pw \cdot G_2$.

The client first generates a random integer $r_c \in (Z/qZ)^*$ and calculates $Y_c = r_c \cdot G_1 + pw \cdot G_2$, where pw is the mutual password of the client. G_1 and G_2 are two base points on the elliptic curve E. After the client accomplishes the above computation, it delivers Y_c to the server.

Step 2. $Server \longrightarrow Client$: $Y_s = r_s \cdot G_1, H(K_s, Y_c^{'})$.

The server receives Y_c and calculates $Y_c^{'} = Y_c - pw \cdot G_2$ to generate its keying material $K_s = Y_c^{'} \cdot r_s$. And then the server calculates $Y_s = r_s \cdot G_1$ and $H(K_s, Y_c^{'})$, in which r_s is a random integer and pw is the pre-shared password of the server. G_1 is base point on elliptic curve E. $H(K_s, Y_c^{'})$ is a hash function that is based on the server's keying material K_s and $Y_c^{'}$. The server sends $Y_s = r_s \cdot G_1$ and $H(K_s, Y_c^{'})$ to the client.

Step 3. $Client \longrightarrow Server$: $H(K_c, Y_s)$.

The client calculates her/his keying material K_c, where $K_c = Y_s \cdot r_c$, and then the client delivers a hash function $H(K_c, Y_s)$ to the server. Finally, both parties verify $H(K_c, r_c \cdot G_1) = H(K_s, Y_c^{'})$ and $H(K_s, r_s \cdot G_1) = H(K_c, Y_s)$.

After the keying material is verified, in this stage, the session key is generated as follows.

$$SK_c = H(K_c),$$
$$SK_s = H(K_s),$$
$$Session\ Key = SK_c = SK_s.$$

Obviously, our scheme does not involve any pre-determined distinct value when the two parties calculate their key material or session key. There are only 3 steps in our scheme, where elliptic curve multiplications and addition operations are in Step 1 and Step 2, the one-way hash function is in the last step and the session key generation stage. Two parties can easily accomplish the whole authentication process in a rough way (smart card, wireless communication) and keep the session key in a small storage size device.

4. Security Analysis

In this section, we shall demonstrate the security of our scheme by raising and fighting against several possible attacks. In the analysis below, Eve is an adversary. Our security definitions are described as follows

DEFINITION 1. The elliptic curve discrete logarithm problem (ECDLP) is that given a public key point $Q_i = x_i \cdot G$, to compute secret key x_i is hard. For this reason, in our scheme, we can add some parameters as passwords to extend the equation $Y_i = r_c \cdot G_1$, for given Y_i, to obtain r_c and pw is hard.

DEFINITION 2. The one-way hash function $Y = H(X)$ ensures that given Y, to compute X is hard.

Man-in-Middle Attack. The mutual password between the client and the server is used to prevent the man-in-middle attack. The illegal user Eve cannot pretend to be the client or the server to authenticate the other since he/she does not own the mutual password.

Password Guessing Attack. The on-line attack cannot succeed since the server can choose appropriate trail intervals. On the other hand, in the off-line password guessing attack, Eve can try to find out the weak password by repeatedly guessing possible passwords and verifying the correctness of the guess based on obtained information in an off-line manner. In our scheme, Eve can gain the knowledge of $Y_c = r_c \cdot G_1 + pw \cdot G_2$, $Y_s = r_s \cdot G_1$, $H(K_s, Y_c')$ and $H(K_c, Y_s)$ in Steps 1, 2 and 3, respectively. Assume Eve wants to impersonate the client. He/she first guesses password pw' and then finds $Y_c' = Y_c - pw' \cdot G_2$ and Y_s. However, Eve has to break the elliptic curve discrete logarithm problem to find the keying material $K_c = K_s$ to verify his/her guess. Eve cannot gain the session key without Y_c' and the keying material K_c, K_s.

Replay Attack. Eve intercepts $Y_c = r_c \cdot G_1 + pw \cdot G_2$ from the client in Step 1 and uses it to impersonate the client. However, Eve cannot compute a correct $H(K_c, Y_s)$ and deliver it to the server unless he/she can guess pw right to obtain Y_s and guess the right r_s, and then Eve must face the elliptic curve discrete logarithm problem. On the other hand, suppose Eve intercepts $Y_s = r_s \cdot G_1$ and $H(K_s, Y_c')$ from the server in Step 2 and uses it to impersonate the Server. For the same reason, if Eve cannot gain the correct r_c, the client will find out that $H(K_c, r_c \cdot G_1)$ is not equivalent to $H(K_s, Y_c')$, and then the client will not send $H(K_c, Y_s)$ back to Eve.

5. Performance Comparison

In this section, we will compare the computational complexity of our scheme with that of the Pretty-Simple PAKE protocol. To analyze the computational complexity, we first define the following notations.

T_H – the time for computing the adopted one-way hash function $H(\cdot)$.

T_{MUL} – the time for computing modular multiplication.

T_{EXP} – the time for computing modular exponentiation.

T_{EC_MUL} – the time for computing the multiplication of a number and a point on the elliptic curve.

T_{MAC} – the time for computing the adopted $MAC_k(\cdot)$.

The elliptic curve discrete logarithm problem with order 160-bit prime offers approximately the same level of security as the discrete logarithm problem with 1024-bit modulus (Koblitz *et al.*, 2000; Schroeppel *et al.*, 1995). Computing the multiplication of a number and a point on the elliptic curve and a modular exponentiation requires an average of 29 1024-bit and 240 1024-bit modular multiplications, respectively. Thus, T_{EC_MUL} is expected to be about 8 times faster than T_{EXP}; i.e., $8 \times T_{EC\_MUL} = T_{EXP}$.

The computations on both sides are only composed of the multiplication of a number and a point on the elliptic curve and the adopted one-way hash function $H(\cdot)$. Obviously, both parties' computational complexities in our scheme are more efficient than those in the Pretty-Simple PAKE protocol, and the even more considerable computation overhead emerged from the server was in charge of large amount of clients. Assume that there is n users in the system, in our scheme, the server only computes the elliptic curve multiplication operation $n \times (2 \times T_{EC\_MUL})$. On the contrary, the server in the Pretty Simple PAKE scheme computes the modular exponentiation $n \times (4 \times T_{EXP})$. Since the elliptic curve multiplication operation is faster than modular exponentiation and the number of times in our scheme is also less than those in the Pretty Simple PAKE protocol, our scheme mitigates the computation overhead of the server in the a large amount of clients environment.

Moreover, our scheme requires fewer steps to agree on a session key than the Pretty-Simple PAKE protocol, and it only needs low storage devices. In the Pretty-Simple PAKE protocol, the secrecy-amplification phase and the session-key generation phase have to involve not only the mutual password but also 3 pre-determined distinct values Tag_c,

Table 1
Computational complexities of our scheme and Pretty-Simple PAKE protocol

	Our scheme	Pretty-Simple PAKE protocol
The client's computations	$2 \times T_{EC\_MUL} + 3 \times T_H$	$4 \times T_{EXP} + 3 \times T_{MAC} + 2 \times T_{MUL}$
The server's computations	$2 \times T_{EC\_MUL} + 3 \times T_H$	$4 \times T_{EXP} + 3 \times T_{MAC} + 2 \times T_{MUL}$
Steps required	3	4

Tag_s and Tag_{sk}. On the contrary, our scheme only needs a slight storage size to keep the mutual password, and the computational complexity is also lowered so that it can be easily employed in a rough system (wireless network, smart card).

6. Conclusions

Although the Pretty-Simple PAKE protocol has its strengths in robustness and simplicity, it increases system computations and adds additional buerdens on the wireless network environment. In this paper, we reduce the coordination steps between the client and the server and apply elliptic curve cryptography to the protocol. Besides, in our scheme, the client and the server just need to negotiate the password in the registration stage. Since the wireless device desires to reduce the computation load and storage size, our proposal is suitable for the wireless network environment.

Acknowledgements

The authors wish to thank many anonymous referees for their suggestions to improve this paper. Part of this research was supported by the National Science Council, Taiwan, R.O.C., under contract no. NSC91-2626-E-324-001.

References

Diffie, W., and M. Hellman (1976). New direction in cryptography. *IEEE Transactions on Information Theory*, **22**(6), 472–492.

ElGamal, T. (1985). A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, **IT-31**, 469–472.

Goldreich, O., and Y. Lindell (2001). Session-key generation using human passwords only. *Lecture Notes in Computer Science*, **2139**, 408–432.

Katz, J., R. Ostrovsky and M. Yung (2001). Efficient password-authenticated key exchange using human-memorable paswords. *Lecture Notes in Computer Science*, **2045**, 475–494.

Kobara, K., and H. Imai (2002). Pretty-simple password-authenticated key-exchange protocol proven to be secure in the standard model. *IEICE Transactions on Fundamentals*, **E85-A**(10), 2229–2237.

Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, **48**, 203–209.

Koblitz, N., A. Menezes and S.A. Vanstone (2000). The state of elliptic curve cryptography. *Designs, Codes and Cryptography*, **9**(2/3), 173–193.

Menezes, A. (1993). *Elliptic Curve Public Key Cryptosystem*. Kluwer Academic Publishers.

Menezes, A., and S. Vanstone (1995). Elliptic curve systems. *Proposed IEEE P1363 Standard*, 1–42.

Miller, V. (1986). Use of elliptic curves in cryptography. In *Advances in Cryptology*, *CRYPTO'85*. pp. 417–426.

Naor, M., and M. Yung (1989). Universal one-way hash functions and their cryptographic applications. In *Proc. of the 21st STOC*. pp. 33–43.

Rivest, R.L., A. Shamir and L. Adleman (1978). A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, **21**(2), 120–126.

Schroeppel, R., H. Orman, S. O'Malley and O. Spatscheck (1995). Fast key exchang with elliptic curve systems. In *Advances in Cryptology, CRYPTO'95*. pp. 43–56.

T.-Y. Chang received the BS in information management from Chaoyang University of Technology (CYUT), Taichung, Taiwan, Republic of China, in 2001, and his MS in Department and Graduate Institute of Computer Science and Information Engineering from CYUT, in 2003. He is currently pursuing his PhD in computer and information science from National Chiao Tung University, Taiwan, Republic of China. His current research interests include information security, cryptography, and mobile communications.

C.-C. Yang received his BS in industrial education from the National Kaohsiung Normal University, in 1980, and his MS in electronic technology from the Pittsburg State University, in 1986, and his PhD in computer science from the University of North Texas, in 1994. He has been an associate professor in the Dept. of Computer Science and Information Engineering since 1994. His current research interests include network security, mobile computing, and distributed system.

C.-M. Chen received the BS in information management from Kun Shan University of Technology, Tainan, Taiwan, Republic of China, in 2002. He is currently pursuing his MS in the Graduate Institute of Networking and Communication Engineering from Chaoyang University of Technology. His current research interests include information security, network routing, and mobile communications.

Nesudėtingo raktų apsikeitimo su autentifikacijos slaptažodžiu protokolo, skirto bevieliam ryšiui, patobulinimas

Ting-Yi CHANG, Chou-Chen YANG, Chia-Meng CHEN

Straipsnyje pasiūlytas gerai žinomo robastinio ir paprasto raktų apsikeitimo su autentifikacijos slaptažodžiu protokolo PAKE patobulinimas. Pasiūlytoje dar efektyvesnėje protokolo schemoje pakanka iš anksto žinoti tik slaptažodį, be to skaičiavimai yra parastesni. Protokolas remiasi eliptinės kreivės diskretinio logaritmo apskaičiavimo problema, todėl išlošiama dėl rakto dydžio, spartumo ir saugumo. Ši naujoji schema yra efektyvi, sparti ir pigi, todėl ypač tinka suagiam bevieliam ryšiui.