Fast Password Authentication Protocol in Auto-guard of Car Based on Cellphone

Jiandong Meng
Academy of Social Sciences
Shandong Medical College
Linyi, China
cnmengjd@outlook.com

***Abstract*—Due to the shortcoming of existing auto-guard of car, we propose its new design philosophy. First, the car can be controlled and locked by a cellphone, in which the user identity is confirmed by password, and the data are transferred through the Bluetooth or RF channel. Then, we present a secure and fast mutual password authentication protocol. Accordingly, the system architecture of car's auto-guard is designed, which consists of an embedded system in server and a cellphone software in client. Our experiments show that the new proposal is efficient and convenient which can solve most of the existing security problems.**

Keywords-cryptography; password authentication protocol; auto-guard of car; cryptography engineering; Bluetooth

I. INTRODUCTION AND MOTIVATION

Auto-guard of car [1], as one of the authentication devices used in people's lives, is almost essential for the security of every car. It usually consists of remote controller outside and controller inside. According to cryptographic protocol [2], we may just as well call the former "client" and the latter "server". The traditional schematic diagram of the car's auto-guard is described in Figure 1.

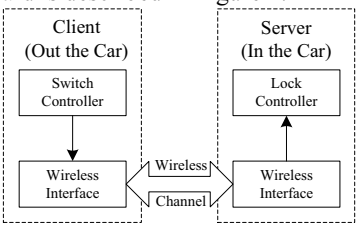

Figure 1. Principle of car's auto-guard

In the research we find that the current auto-guard of car manufactured by firms mainly adopts the security mechanism of pre-distributed symmetric key. It achieves mutual authentication through the symmetric keys fixed in the client [2]. There are many disadvantages in such scheme: the remote controller, simple designed, is short of high-level security guard and easy to reproduce; the client is unique, once the remote controller is lost, the owner has to open the door with the key and then to deal with their own, or he may resort to the professionals to solve the problem; without the password mechanism the built-in key can not be modified; the remote controller is dedicated and inconvenient to take along.

A new design philosophy of the auto-guard car is proposed in this paper: a cellphone is used to control the inside controller of the car and to manage the doors, the user's identity is authenticated by a password, and the data is transferred through the Bluetooth or the RF technique [3]. The advantage of this new design is that: the storage of the key and cryptographic operations are accomplished within the cellphone-embedded SIM (Subscriber Identity Module) card or TPM (Trusted Platform Module) chip, which can guarantee the client's security and ensure that the remote controller will not be copied or suffer from other effective attacks. It can realize multiple-client convenient management by means of rational design of authentication protocol and UI (User Interface). With the use of password mechanism, the password can be modified in time when the password is divulged or the remote controller is borrowed by others. The cellphone is used as the remote controller, which is convenient to take along. Therefore this new scheme can solve the problems in essence.

Password authentication protocol is one of the most simple, convenient and most widely used authentication modes. How to design a password-based mutual authentication protocol is the main problem that the new design philosophy has to solve. Usually there are three ways to protect the security of the passwords: public key cryptography (such as RSA, ElGamal, ECC and so on), symmetric key cryptography (such as AES, DES and so on) and one-way hash function. Among of them, the hash function is widely used because of its good security and smaller computation load.

We might as well borrow ideas from the currently existing homogeneous protocols. Lamport [4] first proposed encrypting the user's password with the safe one-way hash function. Hwang and Yeh [5] proposed a mutual authentication protocol. Although the original author claimed that this scheme could resist the known attacks, Chun et al. [6] pointed out that this protocol was vulnerable to denial of service (DoS) attack. Peyravian and Jeffries [7] also raised a hash-based authentication proposal and declared that it could resist DoS and off-line password guessing attacks. In 2008, Wang et al. [8] proposed a mutual anonymous password authentication scheme and declared that the efficiency was equivalent to the Hwang and others' protocol but it was more secure.

978-1-5090-4840-3/16 $31.00 © 2016 IEEE
DOI 10.1109/CIS.2016.163

We find that due to the application of public key cryptography, the efficiency of Wang's protocol is much lower than the Hwang and others'. Besides, the characteristics of resist-DoS are not very good and the so-called "weak password attack" isn't really inexistent. In this case, an efficient mutual password authentication protocol based on hash function and message authentication code (MAC) is proposed and we also evaluate the security and efficiency in detail.

Based on the new password authentication protocol, we design system architecture of car's cellphone-based auto-guard and also give the specific implementation. Experiments show that the protocol given in this paper, with high efficiency, can resist a variety of known attacks and achieve the security requirement of car's auto-guard perfectly.

II. AN EFFICIENT MUTUAL PASSWORD AUTHENTICATION PROTOCOL

A new mutual password authentication protocol is proposed in this part for the security problem of Wang's scheme [8] and security requirement of password authentication. The new proposal can make up the Wang's security deficiencies, resist various known attack and improve the efficiency greatly.

A. New Scheme

- Initial Conditions and Symbols

S: the server (unique)
C: the client
ID_C: the identity of user *C*（unique）
PW_C: the password of user *C*
H(): strong collision free hash function
R_c, R_s: random number generated by *C*, *S*
SK_C: symmetric key between *C* and *S*
$MAC_{SK\_C}(M)$: the MAC value of *M* with the *SK_C* as the key, the MAC algorithm is also strong collision free
Comma in the message: splice

- User Register

When the user *C* registers at the server *S*, *C* passes $H(PW_C)$, the hash value of the password PW_C, to the server *S*. Then *S* establishes the password verifier, which is shown in Table 1.

TABLE I. PASSWORD VERIFIER

User ID	Password's Hash Value
ID_1	$H(PW_1)$
ID_2	$H(PW_2)$
…	…
ID_n	$H(PW_n)$

Furthermore, *S* generates the symmetric key *SK_C,* imparts the value to *C* and both sides store the key in safety.

- Implementation (as shown in Figure 2)

$C \rightarrow S$: *Authentication Request*
$S \rightarrow C$: R_S
$C \rightarrow S$: $R_C, ID_C, MAC_{SK\_C}(H(PW_C), R_S, R_C, ID_C)$
S verify *C*
$S \rightarrow C$: $MAC_{SK\_C}(R_S, R_C, ID_C)$
C verify *S*

Figure 2. Mutual password authentication protocol

Step 1: *C* submits the authentication request to *S.*

Step 2: *S* sends a random number R_S to *C.*

Step 3: *C* generates a random number R_C, calculates $MAC_{SK\_C}(H(PW_C), R_S, R_C, ID_C)$ with key *SK_C*，and then sends the MAC value to *S* with R_C、ID_C.

Step 4: After receiving the data from *C, S* looks up the password verifier and gets $H(PW_C)$ based on the ID_C. Then *S* calls *SK_C,* calculates the MAC value of $H(PW_C)$, R_S, R_C, ID_C，and then verifies whether this value is the same with the received value. If the same，*S* will authenticates *C* successfully and the protocol can continue；Otherwise *S* will refuse the authentication request from *C.*

Step 5: *S* calls the key *SK_C* to calculate the MAC value of R_S, R_C, ID_C and then sends this value to *C.*

Step 6: *C* calls the key *SK_C* to calculate the MAC value of R_S, R_C, ID_C and verifies whether this value is the same with the received MAC value. If the same, *C* will authenticate *S* successfully and the protocol is accomplished; Otherwise *C* will refuse the authentication request from *S.*

B. Security Analysis

- Denial of Service Attack (DoS Attack)

When an attacker carries out the denial of service attack on the server *S*，*S* can discover the user's illegality in Step 4. *S* doesn't need to do complex calculations before that and just needs to receive and send data three times. Meanwhile the step of verification just needs one look-up table and one calculation of MAC. The R_S can be reused when the authentication failed, so the overhead of random number generation can be completely ignored under the DoS attack. Therefore this protocol can resist DoS attack very well.

- Replay Attack

The preimage of the MAC used to verify the identity contains the random number chosen by the verifier, so the preimage will never repeat as long as the random number doesn't repeat. Namely, ever-used MAC values will arise with a very low probability, which is equal to the probability of the collision in the MAC algorithms. Therefore this protocol can resist the replay attack very well.

- Password Guessing Attack

Password guessing attack can be divided into two: online and offline. The online password guessing attack can be prevented by limiting the logins. The offline attack is still equivalent to the probability of the collision of MAC algorithms and Hash functions. So the protocol can resist the password guessing attack because of the low probability.

- Stealing Verifier Attack

In the new proposal, there are nothing else than the user's identities and the hash values of the password in the verifier stored by *S.* The symmetric key of the MAC algorithms is not stored so the attacker can't generate the right MAC value even if he gets the password verifier, not to mention obtaining any useful information. For the management of the symmetric key, we can usually adopt USB key and other technologies to ensure the storage security.

- Forge Server Attack

If an attacker wants to forge the server *S*, he must call the key *SK_C* to calculate the MAC value of R_S, R_C, ID_C, or attempt to "replay" by means of previous MAC value. Obviously the latter is not possible. Although the preimage of the MAC operation is transported in a public unsafe channel, it's impossible for the attacker to obtain the key *SK_C*. As a result, the attacker can not forge the server to communicate with users.

- Forge Client Attack

Because the replay is infeasible, the attacker can but calculate the MAC value of $H(PW_C)$, R_S, R_C, ID_C to fake *C.* But even he can steal the verifier and get $H(PW_C)$, he will not be able to generate the right MAC value without the key. Consequently it is also impossible to fake the client.

C. Performance Evaluation

In this paper we implement password authentication protocol based on Hash function and MAC code. Due to the absent use of public key, it's far more efficient than the Wang's authentication protocol based on the public key. Because the server doesn't have any large overhead before the server authenticate the client, the proposal in this paper is better to resist the DoS attack than Wang's. In addition, the use of MAC avoids the symmetric encryption algorithms, which improves the overall efficiency greatly.

D. Contrast

In this part, Hwang's scheme proposed in 2002 [5], Peyravian's scheme proposed in 2006 [7], Wang's scheme proposed in 2008 [8] and the new proposal in this paper are compared from the nine aspects of security and efficiency. Details are given in Table 2.

TABLE II. SECURITY AND EFFICIENCY CONTRAST

	Hwang	**Peyravian**	**Wang**	**Our Scheme**
DoS	No	Yes	No	Yes
Replay	No	Yes	Yes	Yes
Password Guest	Yes	Yes	Yes	Yes
Verifier Lost	No	No	Yes	Yes
Forge S	Yes	Yes	Yes	Yes
Forge C	Yes	No	Yes	Yes
Without PKC	Yes	Yes	No	Yes
Hash	Yes	Yes	Yes	Yes
MAC	No	No	No	Yes

As is shown in the table, the scheme proposed in this paper, most effective, meeting the security needs, is optimal. We will discuss the specific implementation and application in the following.

III. SYSTEM ARCHITECTURE AND IMPLEMENTATION

A. System Architecture

Figure 3 shows the architecture of the proposed auto-guard system of car consisting of two parts - a client outside the car which is mainly composed of the keyboard, the cryptographic SoC and the wireless interface, and a server inside the car which includes not only the cryptographic SoC and the wireless interface, but also the interfaces of both car lock and alarm system. And communication protocols between them two can be implemented properly via wireless channels such as Bluetooth or radio frequency.

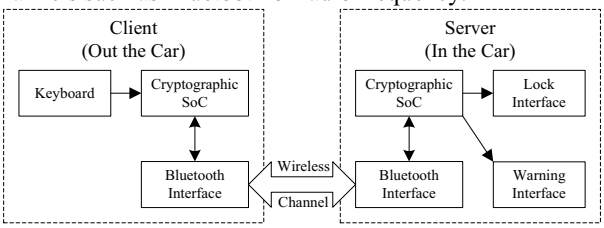

Figure 3. System Architecture

B. Implementation of Client

The most important element in the system architecture is the cryptographic SoC, whose security directly affects the product's performance and functions. And to implement that, using a mobile phone as a client is an excellent choice, since it comes with the keyboard, the Bluetooth interface and even the common smart phone operating system which makes the management and communication of different modules achieved easily.

It should be noted that different from those professional cryptographic devices, the mobile phone operating system and memory card themselves do not have tamper-resistant properties in them. So if one wants to add cryptographic SoC phone features to a cell-phone, the following two methods can be used in practice: SIM card implementation and TPM chip implementation.

Guaranteeing the efficient implementation of password authentication protocols and the relative key storage by the own security and computing capacity of the SIM card is the preferred way for a mobile phone to be added the features of cryptographic SoC. And this can be realized facilely in RF-SIM cards which are commercially available at present. For those SIM cards with computing power, when they begin executing related protocols and then set up the communication with the Server, data can be sent and received by the RF modules embedded in them. Obviously, users who still use the traditional phones without RF-SIM cards can complete the corresponding data transfer though Bluetooth. No matter which way is chosen, the security that even though the data is transferred via an insecure channel, the adversary still can not accomplish any kind of attack can always be ensured by the proposed protocol itself.

Of course, we can also use the phone's built-in TPM chip to store the keys, with reliable software to perform password authentication protocol. This approach makes sure that in the condition that the cryptographic algorithm coprocessor is

absent protocols are actualized securely by using the CPUs embedded in the phones to execute legitimate software programs and then complete cryptographic operations.

We carry this client authentication protocol in Android mobile phone OS [9]. Although the cryptographic algorithms are implemented by software codes in ARM processor, the efficiency of MAC and hash function make the protocol execute in negligible time.

C. Implementation of Server

For implementations of cryptographic SoC in the server, embedded system should be employed. Accordingly, a cryptographic accelerator can be designed to increase efficiency of system. Because of the requirement of password authentication protocol, it includes a hash function, a MAC algorithm, and some other modules.

IP bridge technology is applied to implement the accelerator, which is described in Figure 4. With the help of it, several algorithms can be integrated together, which can reduce memory consumption and data transport optimally.

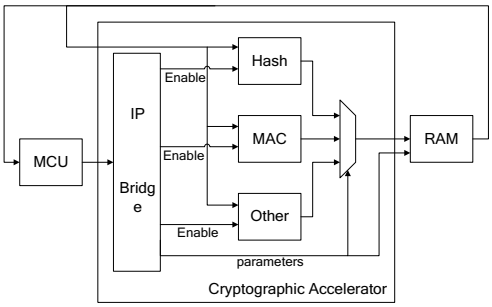

Figure 4. Cryptographic accelerator

Based on the schemes above, we adopt Quartus II to design and implement a cryptographic SoC IP core which has the function of password authentication protocol. In order to make up a practical experiment environment, we combine a FPGA including this IP core with a Bluetooth transceiver, a lock interface, and a warning interface. After analyzing and synthesize the IP core, a report is given by Quartus II 7.2 [10] (in Table 3):

TABLE III. LOGICAL OPTIMUM SYNTHESIS REPORT

Module		Logic Element	Memory	Clock Frequency
8051 MCU		2985	764	33.8
Wireless Interface		486	381	40.2
Random Number Generator		213	160	18.1
Cryptographic Accelerator	MAC	2413	3401	49.1
	SHA-2	1934	607	51.1
	Other	813	148	31.5
IP Bridge		31	18	-

According to a further test, we conclude that client and server can carry out our scheme exactly. That's to say an adversary can't finish any attacks on the auto-guard of car.

IV. CONCLUSIONS

In this paper, we implement an auto-guard of car based on mobile phones and accordingly design a potent two-way password authentication protocol, which can be shown in the results of simulation and several tests of various tools to reach expectative security and overall high efficiency. And this scheme can also solve the problems of similar products like the oversimplification of remote controllers' design, the ease for being duplicated, the inability to modify the keys stored in them, the inconvenience to bring with and so on. So it is of great significance and practical value to the development of auto-guard of car in the future.

Next, we will continue to study the various needs of the users of the auto-guard and design special schemes for different users. Moreover, as people are highly dependent on mobile phones, we can design more function-rich and practical products combined other new technologies with handsets to provide with human more convenient services.

REFERENCES

[1] Auto Guard Security Centre Ltd, "Data sheet of auto guard security products," http://www.autoguardcentre-dy13.co.uk, 2010.

[2] A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone, Handbook of applied cryptography. CRC Press, 1997.

[3] A. Bruce, A. Roberto, C. Praphul, M. Daniel, B. Dan, B. Douglas et al., RF & wireless technologies: know it all. Newnes, Oct. 2007.

[4] L. Lamport, "Password authentication with the insecure communication," Communications of the ACM, vol 24, 1981, pp. 770-772.

[5] J. Hwang and T. Yeh, "Improvement on Peyravian-Zunic's password authentication schemes," IEICE Transactions on Communications, vol. E85-B(4), 2002, pp. 823-825

[6] L. Chun and T. Hwang, "A password authentication scheme with secure password updating," Computers & Security, vol. 22(1), 2003, pp. 68-72.

[7] M. Peyravian and C. Jeffries, "Secure remote user access over insecure networks," Computer Communications, vol. 29(5/6), 2006, pp. 660-667.

[8] B. Wang, H. Zhang, Z. Wang, and Y. Wang, "A secure mutual password authentication scheme with user anonymity," Geomatics and Information Science of Wuhan University, Vol. 33(10), 2008, pp. 1073-1075.

[9] Google, "Google projects for Android," http://code.google.com/intl/en/android, 2010.

[10] Altera, "Product Specification: Cyclone FPGA Family Data Sheet," http://www.altera.com, Oct 2003.