Volume 3, Issue 9, September 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Authenticated Key Agreement Protocol Based on Physically Unclonable Function

Ali M. Allam
Electronics, Communications and Computer Department,
Faculty of Engineering, Helwan University, India

Abstract— A secret key is needed for all application that provides a security service. Sensitive information have to be safe when it is transmitted between parties in an ambient world, therefore, keys are safeguarded and securely deployed from compromise. The present paper proposes an approach based on Physical Unclonable Function (PUF) technology for providing strong hardware based authentication technique between two parties and effective key exchange to assure an authenticated secure channel between them. In addition, the author shows how the properties of PUF can be used to securely set up secret authenticated key agreement mechanism between the two parties with low computational power, and follows cryptographic requirements in designing it. The proposed protocol is more efficient (round complexity), secure, and general purpose compared to previously proposed ones.

Keywords— Key agreement protocols, PUF-Based Cryptography, Authentication, Hardware-assisted cryptographic protocols.

I. INTRODUCTION

In an authenticated key agreement protocol (AKAP) there are two parties or more cooperate over an insecure channel to establish a shared secret key, which is derived as a function of information contributed by, or linked with, each of these parties, (ideally) such that no party can predetermine the resulting value. This common secret key is used to set up a secure channel or to provide secrecy of transmitted information. Hence, there are several security properties and design goals inherent in key agreement protocols; the author is trying to implement AKAP which has the following requirements:

Security considered as the feature where no one except the participants can access to the generated key.
Privacy is the feature that must maintain the privacy of the parties, as possible as.
Perfect forward secrecy means that the expose of long-term keys does not expose past session keys.
Key authentication is the feature where the two parties are assured there is no third party is involved or may gain access to a particular secret key
Key confirmation is the feature where one party is assured that a second party actually has possession of a particular secret key.
Efficiency considers as a feature that contain the following:

- Number of message exchanges (*passes*) required between parties;
- Bandwidth required by messages (total number of bits transmitted);
- Complexity of computations by each party (as it affects execution time); and
- Possibility of pre-computation to reduce on-line computational complexity.

Non-repudiation is a feature that provides ensuring that keying material has been exchanged.

In general, cryptographic protocols that alongside assure high efficiency demands as well as strong security necessities are scarce. One recent trend in this regard is to use the potential of hardware components like signature cards [1], one-time programs [2], standard smart cards [3], or even more complex tokens [4]. The authors in [5] proposed a fingerprint based remote authentication scheme, using a mobile device rather than a smart card. Their scheme involves a fingerprint biometric and password to enhance the security level of the system, while the content of the mobile device is revealed. Moreover, they used hashing functions to implement mutual authentication, which is more secure and efficient than other smart card based.

In 2001, Pappu et al. [6] presented the idea of Physical Unclonable Functions. Physical unclonable functions (PUFs) can be used as a cost-effective means to maintain cryptographic key material in an unclonable way. They can be engaged for strong authentication of objects, e.g., tokens, and of persons possessing such tokens, but also for other purposes. PUFs [7, 8, 9] are innovative primitives to derive secrets from complex physical characteristics of integrated circuits (ICs) rather than storing the secrets in digital memory.

It is essential to review the related work conducted in using PUFs in cryptographic protocols, all the researches that have been worked in this area as far as we know fall into two classes:

The first linked to the PUF and its properties and how to design PUF circuits with properties satisfy the requirement of cryptographic services. Second, using the PUF to provide security services in wireless communication, as special purpose

for Radio Frequency Identification (RFID) or Wireless Sensor Network (WSN), without putting in consideration the fullfilness of other cryptographic requirements beyond secrecy But the author did not notice the presence of any research trying to use the PUF as a building block for cryptographic protocol considering the requirements needed to be fulfilled instead of using mathematical problem which consume power and provide delay. Our motivation is to give a link between the two classes. In [10], they have described PUFs and showed PUFs can provide low-cost authentication of ICs and generate volatile keys for both symmetric and asymmetric cryptographic operations. But didn't include a protocol of how the generated key is exchange between parties and also they send challenge-response pairs (CRPs) on the channel. Which make them to delete the used CRP, lead them to implement PUF with large number of CRP, which is large in implementation complexity A simple PUF-based identification/authentication scheme, based on [8], was proposed in [11]. In this scheme the back-end system identifies many challenge-response pairs for each PUF circuit (i.e., each party), and then uses hundreds of challenges at a time to identify and authenticate parties, probabilistically ensuring unique identification [8] [11]. However, the lack of access control provisions in this approach exposes parties identify by adversarial attack. Moreover, parties do not maintain a state and use any randomness in their responses, making them vulnerable to tracking. Since numerous challenges are necessary for single party identification, many communication rounds between the initiator and the responder are necessary, increasing the required identification time and the party's power consumption.

In [12] a novel key management protocol for wireless sensor network is proposed with is secure against node capture attacks. The security of protocol is based on the use of PUFs and their intrinsic properties. By embedding a PUF on a sensor node chip, makes possible it that nodes be identifiable and can't be reproduced physical uniquely [13].

In conventional usage of a PUF as a key generator, only a fixed number of secret bits need to be generated from the PUF. These bits can be used as symmetric key bits or used as a random seed to generate a public/private key pair in a secure processor [14]. However, in order for the PUF outputs to be usable in cryptographic applications, the noisy bits need to be error corrected, with the aid of helper bits, commonly referred to as a syndrome. The greater the environmental variation a PUF is subject to, the greater the possible difference (noise) between a provisioned PUF response and a re-generated response. This conventional method of PUF key generation using PUF response bits as secret keys has been explored in many publications including [15-18]. Error correction has to be secure, robust and efficient. The security concern is the leakage of secret bits through the syndrome or helper bits. In [19] presented a viable method of PUF-based key generation that is notable for low clock latency and hardware requirements: only a PUF, registers, bit-comparison, and threshold computation logic are required. The generation of keys can be made faster and the security level raised by increasing the number of PUFs.

In [20] presented empirical PUF key generation test results in the context of a PUF application-specific integrated circuit (ASIC) implementation with integrated error correction. Specifically, a total of 133 PUF devices comprising seven PUF circuit layout implementations were designed, implemented, and tested. Four metrics specific to PUF key generation were defined, and empirical data was obtained from 0.13 μm ASICs.

According to the author best knowledge that all the PUF- based key agreement protocol published put only the analysis of the ability of PUF to generate noise-free, uniform-distribution key and didn't pay attention to design a key agreement which contain cryptographic properties of key agreement. In addition, there is no PUF-based protocol provides both services – authenticated key exchange and data integrity- in one protocol beside that all protocols in literature are not general purpose i.e orientated to specific application like WSN or RFID. The main purpose of this paper is to use the cryptographic requirements for authenticated key agreement protocol to design *point-to-point* key agreement mechanism between two parties communicating directly. The advantage of the proposed protocol over existed protocols that, the parties didn't send the response over channel so the attacker can't detect the distribution of the response of the specific PUF, beside that using PUF with short length response will be possible to simple the implementation complexity. One of the applications suggested to be oriented to the proposed protocol is smart grid's neighbor area network. The rest of this paper is organized as follows. Section II, provides background material for PUF. Section III, discuss suggested authenticated key agreement protocols, based on PUF technique. Section IV, addresses the analysis of our authenticated key agreement protocol. Section V, contains paper conclusion.

II. PRELIMINARIES

This section gives an overview of the key hardware building block used in the suggested protocol.

A. Physically Uncloneable Functions

A Physically Uncloneable Function is a source of randomness that is implemented by a physical system. Generally, the randomness of PUFs relies on uncontrollable manufacturing variations during their fabrication. For PUF evaluation, the physical system is queried with a stimulus, usually called challenge. The device then produces a physical output, which is usually referred to as response. A pair of a stimulus and an output is called a challenge/response pair (CRP). Furthermore, a PUF, being a physical system, might not necessarily implement a mathematical function, i.e., querying the PUF twice on the same challenge may yield distinct responses. However, designers need such "noise" to be bounded so that the two responses are closely related in terms of distance.

Security of PUFs

In the literature, PUFs security features introduced in [6, 21] such as unpredictability, uncloneability, bounded noise, uncorrelated outputs, one-wayness, and tamper-evidence. The main security properties of PUFs are uncloneability and unpredictability. Unpredictability is covered using an entropy condition on the PUF distribution. This condition also implies mild forms of uncloneability as well as uncorrelated outputs. Moreover, one usually requires that tampering with

PUFs can be detected easily, the idea being that a user does not use the PUF anymore after detecting it has been tampered with.

The actions of the PUF on input a challenge c should be unpredictable, i.e., has some considerable amount of uncertainty, even if the PUF has been measured before on several challenge values. Here, (conditional) min-entropy is a main tool. It indicates the residual min-entropy on a response value for a challenge c, when one has already measured the PUF on (not necessarily different) challenges $c_1, \dots, c_l$ before. Since the random responses are not under adversarial control we can look at the residual entropy for the answer to r by taking the (weighted) average over all possible response values $r_1, \dots, r_l$.

B. PUFs and Fuzzy Extractors

By nature, PUF evaluation is noisy, so that same stimuli results in closely related but different outputs. Designers use fuzzy extractors of Dodis et al. [22] to convert noisy, high-entropy measurements of PUFs into reproducible random values.

A fuzzy extractor consists of a pair of algorithms (*Gen*, *Rep*). The generation algorithm *Gen* takes as input a noisy measurement *w* and generates as output a secret *st* together with helper data *p*. The helper data can be stored publicly, since it does not reveal information about the secret. It is later used to reproduce the same secret *st* from related measurements. That is, the reproduction algorithm *Rep* takes as input a noisy measurement w' and helper data *p*. If *w* and w' are sufficiently close, Rep gives the same reply *st*. The value *st* is distributed almost uniformly and thereby allows to be used for cryptographic purposes.

Response Consistency: The fuzzy extractor helps to map two evaluations of the same PUF to the same random string, i.e., if PUF is measured on challenge *c* twice and returns *r* and r', then for $(st, p) \leftarrow Gen(r)$, one has $st \leftarrow Rep(r', p)$.

III. PROPOSED PROTOCOL

The protocol uses the cryptographic requirements mentioned in the introduction. Spontaneously, the proposed AKAP proceeds as follows. There are two phases of the proposed protocol, an enrollment phase, a server issues a PUF, measures for a set of randomly chosen challenges the corresponding responses, and finally ensures a noisy-free PUF measurement by generating for each response *r* a fuzzy extractor secret *st* from a set of random secrets as well as a corresponding helper data *p*. The server then sends the PUF to the client. Upon finishing the enrollment phase the server broadcasts a randomly chosen challenge *c* including its helper data *p* to the client and reproduces the secret *st* to be used in the protocol. The client evaluates the PUF on the challenge *c*, computes the corresponding fuzzy secret *st* due to the helper data *p*, and obtains the secret *st*. Consequently, both parties use the fuzzy extractor secret *st* to generate the key. To summarize that, server is the party which holds the data base of CRP and client the one who hold the PUF device, we assume that the PUF transfer security from the server to the client and both of them have random generator and secure storage area. Either the server or the client can play the role of initiator or responder. The role which play by server or client select according to the one who need to not reveal its identity.

A. Notation

The followings are notations used in the proposed protocol:

I	Initiator of the protocol.
R	Responder.
c_i, c_r	Challenge string from initiator or responder.
n_i, n_r	Nonce generated by initiator or responder.
p_i, p_r	Helper data string for initiator or responder.
st_i, st_r	Secret string for initiator or responder.
ID_i, ID_r	Initiator identity and responder identity.
$H_k(M)$	Keyed hash (e.g., HMAC) of message ***M*** using key ***k.***

B. Protocol Messages

$I \rightarrow R : c_i, n_i, p_i, ID_i$

$R \rightarrow I : c_r, n_r, p_r, H_k(c_r, n_i, st_i, p_r, ID_r)$

$I \rightarrow R : H_k(c_i, n_r, st_r, p_i, ID_i)$

The suggested protocol involves message exchanges require precise definition of both the messages to be exchanged and the actions to be taken by each party.

Message (1) is straightforward; note that it assumes that the initiator already will be in possession of the data base of CRP related to PUF device with the responder. The initiator selects randomly a challenge c_i from CRP list and its corresponding data helper p_i. This message also contains an indication to *ID* of the initiator. ID_i is sent in the clear; however, the responder's ID in Message (2) is hashed, so there is no loss of privacy for the responder. Nonce $n_i$is used for message freshness and avoids replay attack.

Message (2) is more complex. Assuming that the responder received the message, he replies with c_r, n_r, p_r and compute st_i by using a combination of PUF device and Fuzzy extractor from c_i, p_i then compute the shared key *k* by $st_i \oplus st_r$. Using keyed hash to protect it's ID and an authenticator calculated from a secret, keyed hash is known to the responder; the authenticator is computed over the shared secret key, the nonce, and the response for initiator's challenge. The keyed hash used also for data integrity.

Message (3) echoes back the data sent by the responder, including the authenticator. The authenticator is used by the responder to verify the authenticity of the returned data. The authenticator also confirms that the sender of the Message

(3) used the same ID as in Message (1), this can be used to detect and counter a Denial of Service attack. A valid authenticator indicates to the responder that a roundtrip has been completed (between Messages (1), (2), and (3)).

IV. ANALYSIS OF PROPOSED PROTOCOL

This section overviews the security analysis of the suggested protocol. The traditional security analysis of authenticated key agreement protocols as in [23,24, 25] can't be follow in analyzing the suggested protocol because the proposed protocol didn't relay on hard computational problem as discrete logarithm problem as in [23,24, 25] but on PUF properties. So, the author will follow two main approaches to analyzing the security of the protocol. One is the fulfillness of cryptographic requirements. The other is the cryptographic approach, which accounts for the fact that components used – PUF – are secure. Here, security of protocols is proven based on proprieties of PUF.

The proposed protocol fulfills the following cryptographic requirements that use them when designing the protocol:

- *Security feature*, the protocol message flow shows that the generated key is computed at initiator and the responder without transmitting it. Besides that, the one who has CRP database corresponding to PUF device can only compute the key.
- *Privacy feature* means that the protocol must not reveal the identity of a participant to any unauthorized party, including an active attacker that attempts to act as the peer. Clearly, it is not possible for a protocol to protect both the initiator and the responder against an active attacker; one of the participants must always "go first." In general, we believe that the most appropriate choice is to protect the responder by hashing it's *ID* in message (2) in the suggested protocol, since the responder is typically a relatively anonymous "client," while the initiator's identity may already be known as in case of smart grid.
- *Perfect forward secrecy* (PFS) feature is inherent in the protocol due to depending on physical construction of the PUF not on long term key.
- *Key authentication* and *Key confirmation* are satisfied due to PUF, because the key is generated only from the circuit design in PUF device to get the response to generate the key so there is implicated authentication to generate the correct key. Messages (2 and 3) act as handshaking to complete the authentication.
- The *Efficiency* feature is worth discussing. In many protocols, key setup must be performed frequently enough that it can become a bottleneck to communication. The key exchange protocol must minimize computation as well total bandwidth and round trips. Round trips can be an especially important factor when communicating over unreliable media. Using our protocols, only one round-trip are needed to set up a working security association, the third message for authentication. This is a considerable saving in comparison with existing protocols. Besides that, our protocol depended only PUF device without any computation needed and key setup.
- *Non-repudiation*. No one of the two parties can deny the negotiation to give the key because it is assumed the PUF device is unclonable.

As mentioned above that all the requirements depend on the uncloneability of PUF and its unpredictability.

Firstly, the analysis in [26] presented how difficulty of random duplication of PUF is**.** To create a forged product, an attacker can attempt to fabricate a clone containing a PUF with the exact same type as in the original product. Assume that the attacker also has the design plans of the product, including masks of the IC and the specification of the PUF. The statistical variation in PUF fabrication ensures that for an attacker to successfully create an identical PUF, depending on the PUF's entropy, a large number of ICs need to be fabricated in order to discover a suitable counterfeit.

Assume that the entropy in bits *b*. This means there are potentially 2^b numbers of different PUFs possible. Assume that all the possibilities occur with equal probability. In [26] introduced *k* as the number of valid genuine PUFs the attacker knows, so if a fabricated clone matches one of these *k* PUFs, the attacker can successfully create a counterfeit product with this clone. Let *a* as the number of PUFs the attacker produces (at random), it is possible to bound the number of successfully created cloned PUFs as (based on the birthday collision attack):

$$s = \frac{a \,.\, k}{2^b}$$

If the cost of creating a PUF is expressed as *c* and the profit of a successful counterfeit product as *p*, it is straightforward to calculate when cloning is profitable:

$$p \,.\, s > a \,.\, c = \frac{p \,.\, k}{2^b} > c$$

Which means the required entropy to make the attack unprofitable is:

$$b > \log k \,.\, \frac{p}{c}$$

This examples illustrates the importance of entropy in PUF responses; the larger this entropy the higher the cost for an adversary to find a successful clone.

Secondly, the main measure for determining the uncertainty or unpredictability of a PUF is entropy. In the PUF context, it is a measure of uncertainty about an unknown response.

Let *x* be the PUF challenge for which the adversary should predict the response *y*. Further, let *Y (x)* is the random variable representing *y*. Moreover, let *W(x)* is the random variable representing the set of all responses of the PUF except *y*. We are interested in the conditional min-entropy:

$$H_\infty(Y/W) = -log_2(\max_{x \in X}\{\Pr[Y(x)/W(x)]\})$$

Which quantify the minimal number of bits of *y*, which cannot be predicted by the adversary, even in case all other responses in *W(x)* are known. Hence, $2^{-H\infty(Y/W)}$ is an information-theoretic upper bound for the probability that an adversary guesses the PUF response *y* to challenge *x*.

As shown from the protocol only the challenge from both parties is transmitted clear, and the response is hashed. So, whatever the distribution of the challenge:

$$\Pr[Y(x)/W(x)] = zero$$
$$\therefore H_{\infty}(Y/W) = zero$$

As shown, the protocol merit is in using PUF as a building block. The PUF security features provided the proposed protocol with immunity from attacker predicting the secret key.

V. Conclusion

In this paper, an authenticated key agreement protocol is proposed for its use in point to point communication. The security of the proposed scheme is built on PUF security properties. The advantages of the new key agreement scheme include security and efficiency. Besides that, the merit of the proposed protocol is built in devices and need no computational capability which is suitable for the application, which needs low power consumption. The proposed protocol takes advantage of PUF properties to provide secrecy and authenticity of the deployed key against adversaries. The protocol is simple (less communication complexity) and required less additional hardware. In addition, because of the use of PUFs, the protocol provides tamper evidence and uncloneability.

References

[1] D. Hofheinz, D. Unruh, and J. Muller-Quade," Universally composable zero-knowledge arguments and commitments from signature cards," In *In Proc. of the 5th Central European Conference on Cryptology MoraviaCrypt 2005*, 2005.

[2] S. Goldwasser, Y. Kalai, and G. Rothblum," One-time programs," *Lecture Notes in Computer Science*, vol 5157, pp. 39-56, 2008.

[3] C. Hazay and Y. Lindell, "Constructions of truly practical secure protocols using standard smartcards," In *ACM Conference on Computer and Communications Security*, pp. 491-500, 2008.

[4] J. Katz," Universally composable multi-party computation using tamper-proof hardware," In *Moni Naor, editor, EUROCRYPT*, vol. 4515 of Lecture Notes in Computer Science, pp. 115-128. Springer, 2007.

[5] Chin-Ling Chen, Cheng-Chi Lee and Chao-Yung Hsu," Mobile device integration of a fingerprint biometric remote authentication scheme," *International Journal of Communication Systems*, vol.25, no.5, pp. 553–688, May 2012.

[6] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

[7] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Controlled physical random functions," In *Proceedings of 18th Annual Computer Security Applications Conference*, Dec. 2002.

[8] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," In *Proceedings of the Computer and Communication Security Conference*, Nov. 2002.

[9] J. Lee, D. Lim, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits with identification and authentication applications," In *Proceedings of the IEEE VLSI Circuits Symposium*, June 2004.

[10] G. Suh, and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation", *ACM*, 2007.

[11] D. Ranasinghe, D. Engels, and P. Cole, "Security and Privacy: Modest Proposals for Low-Cost RFID Systems," *Proc. Auto-ID Labs Research Workshop*, Zurich, Switzerland, September 2004.

[12] R. Bahrampour and R. Atani, "A Novel Key Management Protocol for Wireless Sensor Networks Based on PUFs," *International Journal of Future Generation Communication and Networking*, vol. 6,no. 2, Apr. 2013.

[13] J. Guajardo, S. Kumar and P. Tuyls, "Key Distribution for Wireless Sensor Networks and Physical Unclonable Functions," *Secure Component and System Identification Workshop - SECSI Berlin Germany*, 2008.

[14] G. E. Suh, "*AEGIS: A Single-Chip Secure Processor*," Ph.D. thesis, Massachusetts Institute of Technology, Aug. 2005.

[15] B. Gassend, "*Physical Random Functions*," Master thesis, Massachusetts Institute of Technology, Jan. 2003.

[16] C. Bosch, J. Guajardo, A. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient helper data key extractor on FPGAs," In *Cryptographic Hardware and Embedded Systems (CHES)*, pp. 181–197, 2008.

[17] R. Maes, P. Tuyls, and I. Verbauwhede, "Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs," In *Cryptographic Hardware and Embedded Systems (CHES)*, pp. 332–347, 2009.

[18] M. Yu and S. Devadas, "Secure and robust error correction for physical unclonable functions," *IEEE Design and Test of Computers*, vol. 27, pp. 48–65, 2010.

[19] Paral, Zdenek, and S.Devadas, "Reliable and Efficient PUF-based Key Generation Using Pattern Matching," In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 128–133,2011.

[20] Yu, Meng-Day et al., "Performance Metrics and Empirical Results of a PUF Cryptographic Key Generation ASIC," In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 108–115, 2012.

[21] Su, Holleman, and Otis," A digital 1.6 pJ/bit chip identification circuit using process variations," *IEEE Journal of Solid-State Circuits*,vol.43, no.1,pp. 69–77 ,Jan. 2008.

[22] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," In *Proc. Advances in Cryptology— Eurocrypt '04*, 2004.

[23] Chuang, Yun-Hsin, Tseng, and Yuh-Min," Towards generalized ID-based user authentication for mobile multi-server environment," *International Journal of Communication Systems*, vol. 25, no. 4, pp. 447–460, Apr. 2012.

[24] Debiao, Jianhua, and Jin," A pairing-free certificateless authenticated key agreement protocol," *International Journal of Communication Systems*, vol.25, no. 2, pp. 221–230, Feb. 2012.

[25] Xie, and Qi," A new authenticated key agreement for session initiation protocol," *International Journal of Communication Systems*, vol.25, no.1, pp.47–54, Jan. 2012.

[26] B. Škori´c," *Physical aspects of digital security*," 2011, Lecture notes.

