HPMAP: A Hash-Based Privacy-Preserving Mutual Authentication Protocol for Passive IoT Devices Using Self-Powered Timers

M. H. Afifi*, Liang Zhou†, Shantanu Chakrabartty† and Jian Ren*

* Department of Electrical and Computer Engineering
Michigan State University,
East Lansing, MI, U.S.A
Email: {afifi, renjian}@msu.edu.

† Department of Electrical and Systems Engineering
Washington University in St. Louis.,
St. Louis, MO, U.S.A.
Email: {liang.zhou, shantanu}@wustl.edu.

***Abstract*—The proliferation of passive Internet-of-Things (IoT) into the consumer and the enterprise market has necessitated enhanced security requirements. Many security protocols have been proposed in literature to address these requirements, however, they are either prone to certain types of attacks or are computationally expensive for resource-constrained passive IoT devices. In this paper we propose two variants of a novel mutual authentication protocol that utilizes the synchronization property of Fowler Nordheim (FN) tunneling based self-powered timers. The first protocol provides mutual authentication using the dynamic timer values. The protocol is both lightweight and provably immune to most of the well-known security attacks. Moreover, it offers an efficient and secure capability for easy revocation of tags and readers from the IoT system. The second protocol, an enhanced version of the first, provides disguised identities for applications that require privacy preserving. This protocol can thus serve as a perfect candidate for high-security passive IoT applications such as e-passports.**

***Index Terms*—Internet of Things, mutual authentication, dynamic authentication, low-cost and passive tags, FN tunneling.**

I. Introduction

Internet of Things (IoT) systems have become an essential part of a wide-set of our daily applications such as smart homes, wearables, retails, health-care, and automotive. Passive IoT systems are composed of three main entities: a reader, a tag, and a back-end server. While readers and servers are assumed to be rich in computational and storage resources, passive tags on the other hand have limited resources due to size and power constraints. Passive tags have no internal power source, instead they rely on the electromagnetic waves harvested from the reader to provide sufficient power for both computations and data transmission. The elimination of the power source makes these passive devices low-cost, light-weight and support long-term lifetime. Hence they are attractive for ubiquitous implementation of IoT systems.

Unfortunately, passive IoT systems are vulnerable to all kinds of security attacks in a Radio-Frequency (RF) channel. Unlike most RF systems, passive IoT systems are not able to incorporate complex encryption techniques due to their limited resources. Also, the incorporation of IoT systems in some sensitive applications such as health-care and e-passports [1] has brought about more advanced security concerns. Mitigating these concerns requires the use of security primitives such as *dynamic mutual authentication*, *easy revocation of tags and readers*, and *privacy preservation*. These high-level requirements serve as the guidelines for our proposed authentication protocols.

In mutual authentication, not only the tag authenticates itself to the reader but also the reader authenticates itself to the tag. More specifically, malicious tags should be recognized by legitimate readers and legitimate tags should provide assurance that malicious readers are not able to access their sensitive information. This eliminates a manifold possible attacks performed by malicious readers. Moreover, the server should preserve all the rights to be able to promptly revoke a tag or a reader at any time t_i for any reason such as detecting a malicious behavior. In particular, the server should assure that the revoked reader will have no advantage of the previously communicated messages to be able to access a tag in the future. Similarly, if a tag is revoked from a system, it should have no advantage of the previously communicated messages to convince the reader of its authenticity. Privacy preservation is a crucial issue in applications that deal with individual's data. Therefore, identity of the tags should neither be communicated in clear-text nor traced by an adversary. A privacy preserving IoT system should always disguise the tag identification from all possible adversaries. Fulfillment of these security aspects ensures the prevention of sensitive information on the tag and the reader from malicious access.

In this paper we propose two variants of a novel mutual authentication protocol that utilizes synchronous self-powered timers to tackle the aforementioned security issues. In particular, a pair of independent timers is incorporated on-board of a tag. These timers are synchronized with their gold-standard counterparts at the reader's side. The dynamic values of these timers are securely communicated between the tag and the reader to provide the desired mutual authentication. With only two hash function computations at the tag's side, the proposed protocol is both lightweight and provably immune to most of the well-known security attacks. Moreover, it provides an efficient and secure capability for easy revocation of tags and

This work was supported in part by research grants from the National Science Foundation (CNS:1525476, ECCS:1550096) and by research contracts from Semiconductor Research Corporation (Contract 2015-TS-2639 and Contract 2015-TS-2640).

978-1-5386-3180-5/18/$31.00 ©2018 IEEE

readers from the IoT system. The second protocol dynamically masks the tag identification in order to provide the desired privacy preserving. This comes at the cost of only one added XOR operation at the tag's side.

The rest of this paper is arranged as follows. Section II presents some related work. Section III briefly describes the principle of operation of the self-powered timers. Our mutual authentication protocol is proposed in Section IV. The privacy preserving variant of the proposed protocol is introduced in Section V. In Section VI, security and performance of the proposed protocols are analyzed. Section VII provides the experimental results. We conclude in Section VIII.

II. Related Work

In order to protect IoT systems from different attacks, many authentication protocols have been proposed to meet different security requirements. In this section, to get an idea of how they overcome different attacks, we provide an overview of some state-of-the-art hash-based authentication protocols.

In a first attempt to achieve mutual authentication between tag and reader, Hash-Lock protocol was proposed in [2]. To achieve privacy, instead of using the tag's ID, this protocol uses the pseudonym of the tag, metaID. However, since the secret key and the ID are sent in clear-text, an attacker can eavesdrop the key and the tag can later be impersonated. To avoid the these limitations, a randomized version of the Hash-Lock protocol was proposed in [3]. In this protocol tags respond to reader's queries by a random value which is then concatenated with the hash of the ID and sent to the reader. The reader searches its database for the ID that corresponds to the hash value which is then sent to the tag in clear-text. While the tag's response varies in each session, it is easy for an adversary to eavesdrop and obtain the identity of the tag. A hash-chain protocol was proposed in [4] to provide better protection for the users privacy. The tag always replies to the reader queries with different responses by depending on two different hash functions. Although this protocol introduces the dynamic tag responses, an attacker can disguise a legitimate tag by resending an intercepted authentication message to the reader. Therefore, the protocol is vulnerable to replay attacks.

A challenge-response approach was used in [5] to propose a Low-cost Authentication Protocol (LCAP). This protocol provides mutual authentication and avoids both desynchronization and traceability attacks. However, it fails to provide forward security. In [6], a hash function, a pseudo-random number generator, and an XOR operator are used in a privacy preserving authentication protocol for low-cost tags. However, as shown in [7], this protocol is vulnerable to replay and denial of service attacks. In [7], an anti-desynchronization privacy preserving authentication protocol was proposed. In this protocol, the server keeps track of the updated random key to prevent the active attackers from de-synchronizing the shared secret between the tag and the server. Although this technique prevents the replay attack, it is prone to denial of service attacks. Finally, in [8], an authentication scheme that utilizes symmetric key cryptography, random number generators, and hash functions was proposed. In this scheme, although the random number generation makes it difficult to predict the next random value, it is susceptible to reverse engineering due to the static structure of the seed.

III. The Self-Powered Timers Operation

The proposed protocols rely on the self-powered FN tunneling timers reported in [9]. The timers are thermodynamically driven and do not require any additional source of powering. Also, the timers have been shown to be synchronized to each other, even if they are spatially separated from each other. The mathematical model describing the dynamic response of the timer is given as:

$$V_i = K_2 / \ln(K_1 t_i + K_0) + K_3, \tag{1}$$

where V_i is the value of the timer at time instant t_i and (K_0, K_1, K_2, K_3) are the model parameters which are determined by the device form factors and its initialization conditions. The behavior of the timer can be controlled by modulating the parameters K_0 and K_1. As indicated in [9], K_0 is determined by the initial conditions and K_1 depends on form factors such as the gate capacitance and tunneling junction area. This is key to our proposed protocol because we will exploit the synchronization performance across timers with identical structure on different chips as well as the unique response of timers with different form factors.

To verify the performance, we exploited the features of three variants of the timers with different form factors. Fig. 1(a) shows the micro-photograph of the three timer devices that were fabricated on a standard silicon process and have a form factor around 300μm $\times$ 600μm. The general structure of each timer is also plotted in Fig. 1(a). The three timers possess identical structure except the capacitor which affects K_1. The values of the capacitances are 4pF, 8pF and 16pF respectively. The temporal responses of the fabricated timers were measured and shown Fig. 1(b). The three timers show distinct response from each other because of the difference in gate capacitance. As indicated by equation 1, a larger gate capacitance implies a smaller K_1, therefore showing a slower voltage reduction rate, which is verified by Fig. 1(b).

The synchronization performance of the timer is a key to the successful implementation of the protocol. The authentication is only successful in the scenario where the identical timer structures implemented on tag and server are always maintained synchronized to each other. Fig. 1(c) shows the temporal response of three timers with identical structure on different dies, and the responses are almost identical to each other. This verifies that the timer can maintain good synchronization across different chips. The robustness of the synchronization attribute of the self-powered timer was further verified in [9]. For timers with different form factors and operating conditions such as temperature, the overall synchronization performance is measured to be better than 40 dB. Extrapolation study was conducted to verify that the timer can operate as long as 3 years, which is vigorous for passive IoT devices.

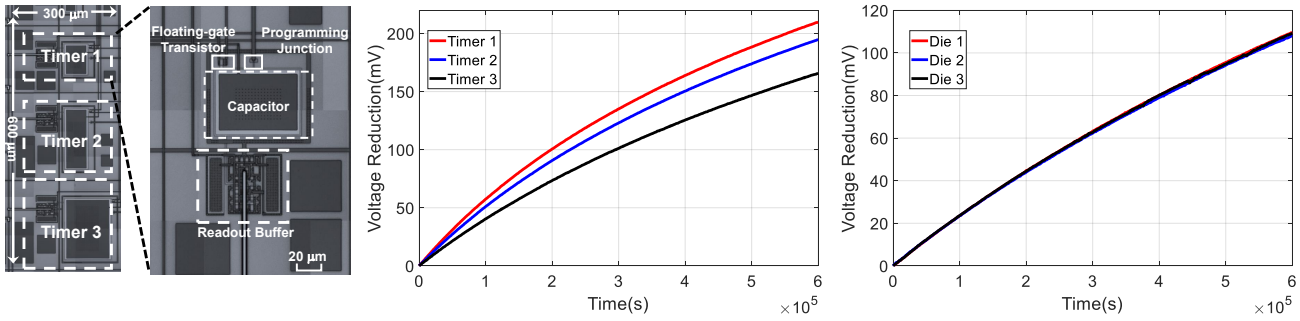

Fig. 1: (a) Die photograph of the three implemented timers with capacitance of 4pF, 8pF and 16pF respectively. (b) Measured response of the three timers. (c) Measured of the timer response at different dies with identical structure.

IV. The Proposed Hash-Based Mutual Authentication Protocol (H-MAP)

IoT systems mostly operate in vulnerable environments. Not only malicious tags attempt to deceive legitimate readers but also the tags are usually susceptible to be accessed by malicious readers. To avoid such acts, in the proposed protocol, the tag comprises two independent self-powered timers: $Timer_1$ and $Timer_2$. As previously shown, the utilized timers dynamically modify their values. One of them is used to authenticate the reader while the other is used to prove the tag's authenticity. Before tag access, not only the tag has to prove its authenticity but also the reader is required to provide a proof of legitimacy. As shown in Fig. 2, the tag first gets authenticated by the legitimate reader by securely proving the knowledge of the timer's value V_1. After authenticating the tag, the reader securely proves the knowledge of the self-powered timer's value V_2. If the tag is convinced about the legitimacy of the reader, full access to its data is granted. The protocol comprises the three following phases.

Initialization: The tag and the server initially share a secret key K at the registration phase.

Tag Authentication: At any authentication time instance t_i the reader sends a request to access the tag. To get authenticated by the reader, the tag computes

$$A_{1,i} = h(K, V_{1,i}),$$

where $V_{1,i}$ is the value of $Timer_1$ at time t_i. The tag replies with the pair $(\mathsf{IDT}, A_{1,i})$. The legitimate reader forwards this pair to the server which is able to retrieve the tag information $(K, \tilde{V}_{1,i}, V_{2,i})$ corresponding to IDT from the server, where $\tilde{V}_{1,i}$ is the value of the corresponding timer at the server side. The server computes $\tilde{A}_{1,i} = h(K, \tilde{V}_{1,i})$, and verifies the authenticity of the tag by checking

$$A_{1,i} \stackrel{?}{=} \tilde{A}_{1,i}.$$

If this does not hold true, the tag is unauthenticated. Otherwise, tag is authenticated and proceeds to authenticate the reader.

Reader Authentication: To achieve mutual authentication, the server computes

$$A_{2,i} = h(K, V_{2,i}),$$

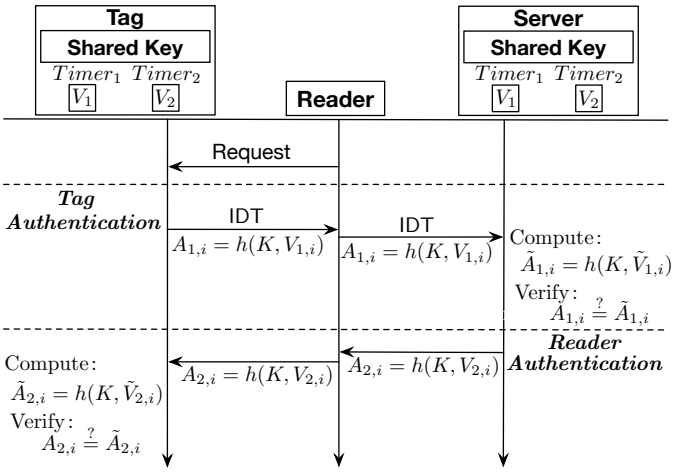

Fig. 2: The proposed mutual authentication protocol H-MAP.

where $V_{2,i}$ is the value of $Timer_2$ at time instance t_i. $A_{2,i}$ is sent to the reader which forwards it to the tag. To authenticate the reader, the tag computes $\tilde{A}_{2,i} = h(K, \tilde{V}_{2,i})$, where $\tilde{V}_{2,i}$ is the value of the corresponding timer at the tag side. The tag verifies the authenticity of the reader by checking

$$A_{2,i} \stackrel{?}{=} \tilde{A}_{2,i}.$$

If the inequality does not hold true, the reader is unauthenticated. Otherwise, the reader is authenticated. Details of the H-MAP are described in Fig. 2. We note that the proposed protocol gives the flexibility to the IoT system to require the mutual authentication or not depending on the application.

Easy Revocation of Tags and Readers: Tags and readers revocation is a trivial setting in any IoT system. We believe that an authentication protocol should enable the server to easily revoke any tag or reader without having any security concern. The server should guarantee that the communicated messages throughout previous authentication sessions shall not give any advantage to a revoked tag or reader over any regular attacker.

Assuming that a server decides to revoke a tag at time t_{rev}. In H-MAP, a revoked tag possesses the values $(K, V_{1,i}, V_{2,i})$ for $i = 0, \ldots, T_{Life}$, where T_{Life} is the tag's lifetime. At any time instance $t_i \geq t_{rev}$, a revoked malicious tag attempts to get authenticated by sending the pair $(\mathsf{IDT}, A_{1,i})$. The server will simply recognize that the tag IDT has been

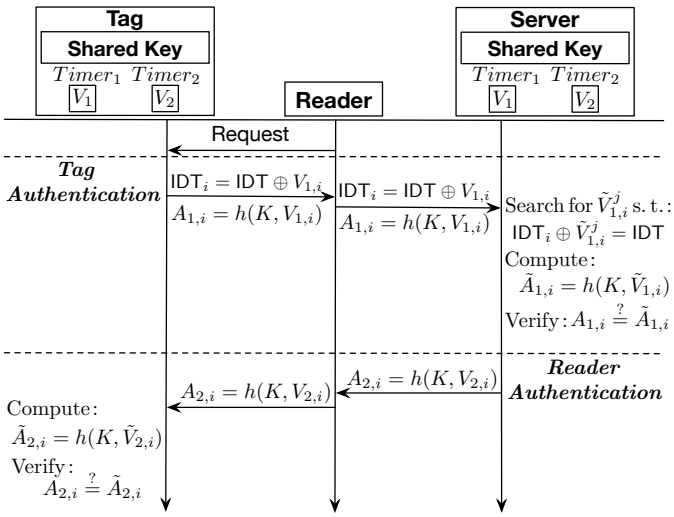

Fig. 3: The proposed privacy-preserving mutual authentication protocol (HPMAP).

revoked from the system. Therefore the values possessed by the revoked tag do not give her any advantage over a regular malicious tag in attempting to get illegitimately authenticated. Similarly, assuming that a server decides to revoke a reader at time t_{rev}. For any targeted tag, an extremely lucky revoked reader possesses all the values $A_{1,i}$ and $A_{2,i}$ for all $t < t_i$. At any $t_i \geq t_{rev}$, a revoked malicious reader will fail to maliciously convince a tag of her authenticity where all the possessed values at $t < t_i$ do not give the revoked reader any advantage to guess the authentication value $A_{2,i}$ over the guess of any regular attacker. Therefore a revoked reader fails to maliciously access a tag at any $t_i \geq t_{rev}$.

V. Privacy-Preserving Mutual Authentication

In this section we introduce the Hash-based Privacy-preserving Mutual Authentication Protocol (HPMAP). The protocol comprises the three following phases.

Initialization: The tag and the server initially share a secret key K at the registration phase.

Tag Authentication: At any authentication time instance t_i the reader sends a request to access the tag. For identification, instead of sending IDT, the tag computes the XOR operation of IDT and the $Timer_1$'s value $V_{1,i}$: $\mathsf{IDT}'_i = \mathsf{IDT} \oplus V_{1,i}$. Also, to get authenticated by the reader, the tag computes

$$A_{1,i} = h(K, V_{1,i}),$$

where $V_{1,i}$ is the value of $Timer_1$ at time t_i. The tag replies to the reader with $(\mathsf{IDT}'_i,\ A_{1,i})$. The reader forwards this pair to the server which searches its database for $\tilde{V}^j_{1,i}$ such that:

$$\mathsf{IDT}'_i \oplus \tilde{V}^j_{1,i} = \mathsf{IDT} \oplus V_{1,i} \oplus \tilde{V}^j_{1,i} = \mathsf{IDT},$$

where $j = 1, \cdots, N$ and N is the number of tags registered on the server's database. The server is then able to retrieve the tag information $(K, \tilde{V}_{1,i}, V_{2,i})$ corresponding to IDT'_i. The server computes $\tilde{A}_{1,i} = h(K, \tilde{V}_{1,i})$, where $\tilde{V}_{1,i}$ is the value of the corresponding timer at the server side. The server then verifies the authenticity of the tag by checking

$$A_{1,i} \stackrel{?}{=} \tilde{A}_{1,i}.$$

If this does not hold true, the tag is unauthenticated. Otherwise, tag is authenticated and proceeds to authenticate the reader.

Reader Authentication: To achieve mutual authentication, the server computes

$$A_{2,i} = h(K, V_{2,i}),$$

where $V_{2,i}$ is the value of $Timer_2$ at time instance t_i. $A_{2,i}$ is sent to the reader which forwards it to the tag. To authenticate the reader, the tag computes $\tilde{A}_{2,i} = h(K, \tilde{V}_{2,i})$, where $\tilde{V}_{2,i}$ is the value of the corresponding timer at the tag side. The tag verifies the authenticity of the reader by checking

$$A_{2,i} \stackrel{?}{=} \tilde{A}_{2,i}.$$

If the inequality does not hold true, the reader is unauthenticated. Otherwise, the reader is authenticated. Details of the HPMAP are described in Fig. 3. We note that HPMAP also provides easy revocation similar to the analysis in Section IV.

In the worst case scenario the server will perform N XOR operations to find the IDT corresponding to the pair $(\mathsf{IDT}'_i,\ A_{1,i})$. The N computations hold true if the system is ideal. However, due to the possible error, latency, or miss-synchronization, we recommend that, for each tag, the server saves m authentication values centered at t_i. Although this requires more work at the server's side, it guarantees a better performance by substantially minimizing the probability of false negatives. Thus, practically, the server is required to perform $m * N$ XOR operations at the worst case scenario to find the IDT corresponding to the pair $(\mathsf{IDT}'_i,\ A_{1,i})$.

In case of large scale systems where a server finds these operations computationally time consuming, a further recommended approach is to divide the database into L groups. During a registration phase, each tag is randomly assigned to a group GID_l where $l = 1, \ldots, L$. During the identification, the tag concatenates the group GID_l to the transmitted pair $(\mathsf{IDT}'_i,\ A_{1,i})$. The server is now required to only perform $m{*}N/L$ operations rather than $m{*}N$. This minimizes the work required by a factor of L. The server can then accordingly design the system parameters (m, L) based on the available storage and computational resources.

VI. Security and Performance Analysis

Security Analysis: The proposed mutual authentication protocols rely on the security of the cryptographic hash function. Specifically, a hash function is said to be cryptographically secure if it satisfies the *Preimage*, *Weak Collision* and *Strong Collision-Resistances*. The first implies that it should be computationally infeasible to find any input for any pre-specified output which hashes to that output, i.e. for any given y, it should be computationally infeasible to find an x such that $h(x) = y$. The second implies that for any given x, it should be computationally infeasible to find $x' \neq x$ such that $h(x') = h(x)$ [10]. Finally, the third implies that it should be computationally infeasible to find any two distinct inputs x and x', such that $h(x) = h(x')$ [11].

In H-MAP, for any authentication attempt at time instance t_i, the exchanged messages between the tag and the reader are $A_{1,i} = h(K, V_{1,i})$ and $A_{2,i} = h(K, V_{2,i})$. Both of the

hash functions arguments $(K, V_{1,i})$ and $(K, V_{2,i})$ are secure. More specifically, K is a private key that is never exposed to the adversary in clear-text and is computationally infeasible to derive. $V_{1,i}$ and $V_{2,i}$ are dynamically and continuously updated with the fresh bits output from the self-powered timers leading to an unpredictable authentication value. Thus, it is infeasible for an adversary, by any means other than exhaustive search, to guess an authentication value, even by overhearing the transmission channel between the tag and the reader. In particular, the adversary can guess a correct a-bits authentication value $A'_i = A_i$ with probability,

$$Pr[(A'_i = A_i)] = 2^{-a}.$$

Therefore, under the assumption that the underlying hash functions have the previously explained characteristics, the proposed protocol is as secure as the cryptographic hash.

In HPMAP, for any authentication attempt at time instance t_i, the exchanged messages between the tag and the reader are $\mathsf{IDT}'_i = (\mathsf{IDT} \oplus V_{1,i})$, $A_{1,i} = h(K, V_{1,i})$, and $A_{2,i} = h(K, V_{2,i})$. An adversary which is able to intercept this tuple for all $t < t_i$ has two possible approaches to attack the security of this protocol. The first is to just guess the arguments of the intercepted hash functions in an attempt to derive the key and the authentication values. Similar to the analysis in HMAP, this approach is practically infeasible. The other approach is to XOR the intercepted IDT'_is together. This reveals the XOR of all the timer values. Nevertheless, since the timer values are random, the adversary gets no advantage of the XORing result to guess any of the timer values at any time $0 < t < T_{Life}$. In other words, the result of $\mathsf{IDT}'_i \oplus \mathsf{IDT}'_{i-1} = V_{1,i} \oplus V_{1,i-1}$, neither gives any advantage to the adversary in finding $V_{1,i}$ nor $V_{1,i-1}$. Therefore, the proposed HPMAP is as secure as the cryptographic hash functions.

In our previous work [12], we extensively analyzed the security implications of using such timers in dynamic authentication of tags. We proved that using these timers provides immunity against most kinds of well-known attacks such as replay, de-synchronization and traceability attacks. Table I compares the security ability of the proposed protocol to some state-of-the-art protocols proposed in literature.

Performance Analysis: To evaluate the performance of the proposed protocol we analyze the design from two main aspects: storage and efficiency. Since tags are typically very resource constrained, this analysis is extremely important to evaluate and compare different designs. Generally, the tag is the part of the system with the least storage and power resources. Therefore, since the reader is assumed to be powerful and has sufficient storage, we focus on the tag's resources.

Storage and area overhead: In terms of storage, the tag is only required to store the static key K. The area overhead on chip for the two self-powered timers is in orders of micrometers. Therefore, the tag area overhead as a result of exploiting the pair of self-powered timers is almost negligible.

Efficiency: At the tag side, H-MAC requires only two hash function computations, while HPMAC requires two hash functions and an XOR computation. In terms of communications, both H-MAC and HPMAC require only two communication sessions between the tag and the reader.

In Table II, we present a cost analysis comparison between the proposed protocol and some of the state-of-the-art protocols. We note that even the protocols that provide a close security and privacy functionality to our protocols, achieve this at a remarkably higher computation and communication costs.

VII. Experiment Results

In this section, we spotlight on the performance of the self-powered timers in terms of their ability to reliably provide the aimed functionality. We show that distinct timers with different structures can generate completely random codes while synchronized timers can generate identical codes for authentication. The timer output is used as a dynamic seeding source for random token generation. As shown in Fig. 4(a), the output of the timer is first digitized using an analog-to-digital converter (ADC). The digital codes generated are then used to feed a Pseudo Random Number Generator (PRNG) such as a Linear Feedback Shift Register (LFSR) to produce random tokens. While the seed for the PRNG is dynamically determined by the timer, the pattern of the PRNG will be continuously broken and thus it functions like a true random number generator. In practical implementations, the timer works in self-powering mode and does not require any external power. The ADC and PRNG modules are powered once authentication request is initiated when external power such as RF power is accessible. Although the timer follows the behavior captured by the model shown in equation 1, the deterministic behavior will be masked by the PRNG and further encoded by the hash function, making the prediction of the token infeasible.

We conducted simulations based on measured data from fabricated timer chip to verify the performance. The first experiment was conducted to validate that using timers with different structures can generate completely independent tokens. To do so, we used an ideal 8-bit ADC to digitize the timers' response shown in Fig. 1(b) where three different timers' responses are considered. The digitized code is then used to seed a 16-stage LFSR with 10000 running cycles to generate random codes. Fig. 4(b) shows the normalized codes generated from the three timers. The first observation is that each timer generates nearly uniform distributed tokens across the time-token space without any pattern, indicating that the tokens are random. The second observation is that at each time instance, the codes generated by the three timers are unrelated to each other. As a result, it is impossible to employ the output of one timer to predict the output of other timers. These two observations confirm our claim that we can integrate two distinct timers on the tag with one for tag authentication and the other for reader authentication.

The second experiment was conducted to verify that the timers with identical structure are well synchronized and can generate identical random codes. To do so, we use the responses from Fig. 1(c) which were measured from the same timer on three different dies that are synchronized to each

TABLE I: Security comparison against various attacks

	Weis et al. [3]	Ohkubo et al. [4]	Song et al. [6]	Fu et al. [8]	Zhou et al. [7]	H-MAP	HPMAP
DoS Attack			✓	✓		✓	✓
MITM Attack			✓	✓	✓	✓	✓
Traceability Attack		✓	✓		✓	✓	✓
Replay Attack			✓	✓	✓	✓	✓
De-synchronization Attack					✓	✓	✓
Mutual Authentication	✓		✓	✓	✓	✓	✓
Privacy Preservation		✓	✓	✓	✓		✓

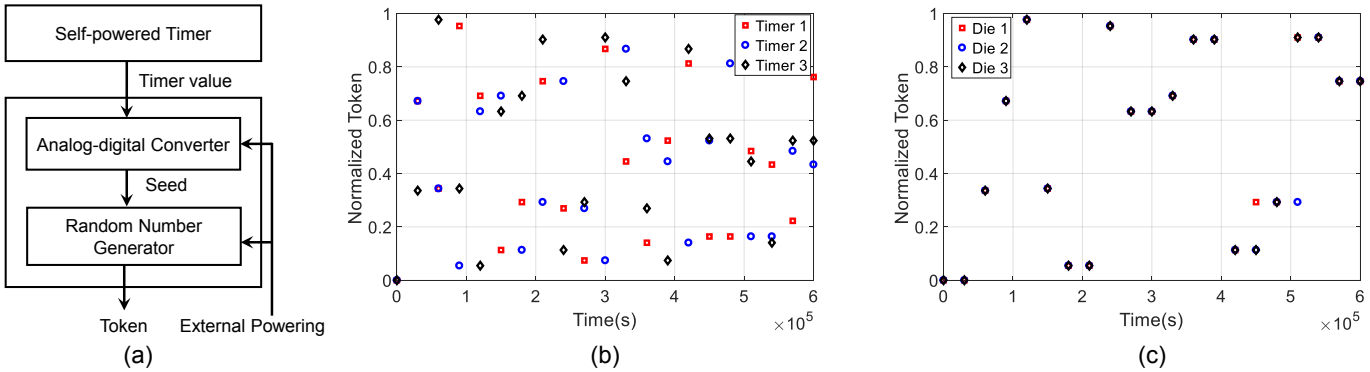

Fig. 4: (a) The method used to generate random tokens using the timer's output. (b) The random tokens generated from timers with different form factors and (c) random tokens generated from identical timers on different dies.

TABLE II: Cost comparison

	# of communications	Computation cost for the Tag
Weis et al. [3]	1	$C_C + C_H + C_R$
Ohkubo et al. [4]	1	$2C_H$
Song et al. [6]	3	$6C_X + 3C_H + C_R$
Fu et al. [8]	4	$5C_C + 2C_X + 4C_H + 2C_R$
Zhou et al. [7]	3	$6C_X + 5C_H$
H-MAP	2	$2C_R + 2C_H$
HPMAP	2	$2C_R + 2C_H + C_X$

C_C: Concatenation cost, C_R: Random number generation cost,
C_X: XOR cost, C_F: Flip operation cost,
C_H: Hash function cost, C_S: Circular shift cost.

other. Using the same settings described in previous paragraph, the normalized codes generated are plotted in Fig. 4(c), where majority of the codes generated at the same time instance are identical to each other except one time instance. This deviation is due to the small mismatch and quantization error, and can be tackled by searching a predetermined range of the reference timer value. This experiment validates that the synchronized timers implemented on the tag and server enable them to generate the identical tokens used for authentication.

VIII. Conclusion

In this paper we proposed a lightweight mutual authentication protocol. The protocol exploited a pair of independent timers incorporated in the tag design. These timers are synchronized with their counterparts at the server side. One timer is responsible for authenticating the reader while the other is responsible for authenticating the tag. This mutual authentication is achieved through comparing the cryptographic hash function of the timer values and a shared secret key. The proposed protocol is proved to be as secure as the underlying cryptographic hash functions. Moreover it provides a reliable lightweight solution to the mutual authentication problem in the IoT devices. An enhanced version of the protocol is proposed for applications that value the privacy preservation at the cost of an added XOR operation.

References

[1] A. Juels, D. Molnar, and D. Wagner, "Security and privacy issues in e-passports," in *Proc. of the First Int. Conf. on Sec. and Priv. for Emerging Areas in Comm. Networks*, (Washington, USA), pp. 74–88, 2005.

[2] S. E. Sarma, S. A. Weis, and D. W. Engels, "Rfid systems and security and privacy implications," *CHES, Springer-Verlag*, pp. 454–469, 2003.

[3] S. Weis, S. Sarma, R. Rivest, and D. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," *Proc. of the 1st Int. Conf. on Security in Pervasive Computing*, pp. 201–212, 2003.

[4] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Hash-chain based forward secure privacy protection scheme for low-cost rfid," *Proc. of the Symposium on Cryptography and Information Security*, pp. 719–724, 2004.

[5] S. Lee, Y. Hwang, D. Lee, and J. Lim, "Efficient authentication for low-cost rfid systems," *LNCS*, pp. 619–627, 2005.

[6] B. Song and C. J. Mitchell, "Rfid authentication protocol for low-cost tags," in *Proceedings of the First ACM Conference on Wireless Network Security*, pp. 140–147, 2008.

[7] S. Zhou, Z. Zhang, Z. Luo, and E. C. Wong, "A lightweight anti-desynchronization rfid authentication protocol," *Information Systems Frontiers*, vol. 12, no. 5, pp. 521–528, 2010.

[8] J. Fu, C. Wu, X. Chen, R. Fan, and L. Ping, "Scalable pseudo random rfid private mutual authentication.," *2nd IEEE International Conference on Computer Engineering and Technology (ICCET)*, pp. 497–500, 2010.

[9] L. Zhou and S. Chakrabartty, "Self-powered timekeeping and synchronization using fowlernordheim tunneling-based floating-gate integrators," *IEEE Transactions on Electron Devices*, pp. 1254–1260, 2017.

[10] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications.," *STOC*, pp. 33–43, 1989.

[11] I. Damgard, "Collision free hash functions and public key signature schemes.," *EUROCRYPT*, pp. 203–216, 1987.

[12] M. H Afifi, L. Zhou, S. Chakrabartty, and J. Ren, "Dynamic authentication protocol using self-powered timers for passive internet of things," vol. PP, 09 2017.