A Strong Mutual Authentication Protocol for SHIELD

Mehmet Hilal ÖZCANHAN[1], Halit TÜRKSÖNMEZ[2]
[1,2]*Department of Computer Engineering, Dokuz Eylul University, Izmir, Turkey*
[1]*hozcanhan@cs.deu.edu.tr*
[2]*halit.turksonmez@ogr.deu.edu.tr*

***Abstract*—Study shows that counterfeit semiconductors or Integrated Circuits (ICs) are increasingly penetrating into advanced electronic defense systems. Traditional supply chain management policies have been found unsuccessful in protecting the IC supply chain. Our study demonstrates that the newly started threat mitigation initiative of Defense Advanced Research Projects Agency's (DARPA) Supply Chain Hardware Integrity for Electronics Defense (SHIELD) scheme has not matured yet, and the proposed authentication protocol improvements are still vulnerable to known non-invasive, side-channel attacks. In present work, a novel authentication protocol based on strong mutual authentication is proposed, which resists the demonstrated attacks on previous schemes. The security and performance comparison with the previous work is provided, to inform the IC community about the seriousness of the weaknesses, in previous works. The comparison results show that our proposed protocol exchanges more information, uses more memory and makes more encryption computations. Thus, although our proposed scheme consumes more energy, it has the security required by SHIELD. The outcome forces IC producers to provide enough memory and processing power in a small die area, if the electronic defense IC supply chain is to have the expected security.**

***Index Terms*—access control, authentication, computer security, integrated circuits, radiofrequency identification.**

I. INTRODUCTION

The new information nourished world has a need for processing, storing and transmission of huge amounts of data. ICs are the main building blocks of digital data collection and processing equipment, as processors and peripheral chips in their central processing units. The need has resulted in a big increase in IC production. Due to flourishing IC manufacturer numbers, companies outsource the fabrication of ICs needed for their end products, to avoid high costs of IC production plants. As a result, the IC supply chain has grown incredibly fast [1]. From production stage to end user delivery, the present supply chain is a global, uncontrollable, multi-hop transfer process spanning over the whole world [2-4]. The IC supply chain consists of designing the ICs through in-house or outsourcing methods, sharing the design with manufacturers, distributing the manufactured ICs to intermediate product manufacturers-device assemblers or globally distributed in-house assembly lines, distributing the products to wholesalers, transferring the products to resellers, delivering the product to the end-users and finally recollecting end of life products. The long supply chain is facing the same problems experienced in other supply chains. Consequently, the copyright, patent and intellectual property (IP) breaches and threats are similarly serious [5]. The U.S. government has formally declared that counterfeit electronics exist in computers, communication devices, aircraft and automobile control circuits, as well as defense systems [6-7]. In a presentation made for public release, DARPA of the U.S. defines the control of the IC supply chain as a "critical problem" [4]. According to a different paper, expired or counterfeit electronics in embedded systems cause the semiconductor industry to lose 4 billion dollars annually [3]. A detailed taxonomy of counterfeit ICs is available [5]. But, counterfeiting has become alarming, after countries have spotted vulnerabilities in their electronic defense systems. A lab report confirms that "IC backdoors" exist in nuclear power plants and weapons control systems [8]. A submitted military thesis details on the counterfeit IC circulation and the amount of profits lost in different electronic supply chains [9]. The thesis also points to the outcome of the 2011 U.S. Senate Armed Services Committee investigation. According to the investigation, counterfeit electronic parts found their way into the advanced systems of the U.S. Department of Defense (DOD) programs [9]. Another work exposed the existence of inexpensive techniques that reintroduce counterfeit ICs into the DoD electronic systems [10]. As a conclusion, it is declared that counterfeit IC based systems put the missions, lives of servicemen in danger and traditional approaches in supply chain risk management are inadequate [2], [9]. The focus on cost and performance is criticized, since it leaves security as an afterthought [11].

To stop the further spreading of the supply chain problems, DARPA initiated a program for eliminating counterfeit ICs from the defense electronics supply chain [12]. The aim of the effort is to make critical systems more resilient to attacks and make counterfeiting too complex and time-consuming to be cost effective [11].

In the rest of this paper, the IC supply chain vulnerabilities, threat models, related countermeasure efforts and their vulnerabilities are explained in Section II. Our motivation, contributions and alternative authentication protocol for SHIELD are presented, in Section III. In Section IV and V, the security and performance analysis, as well as comparison to the criticized works are provided. Conclusion is made in Section VI.

II. RELATED WORK

A. Background Information

The IC supply chain and its vulnerabilities are studied in detail from design to end of life stage, in the literature [1], [13]. After identifying vulnerabilities and threats,

Digital Object Identifier 10.4316/AECE.2020.04010

developing countermeasures gathered momentum among researchers. The threat models that aim at gaining unfair advantages and benefits in the IC supply chain are summarized below [1], [5], [11]:

1. Hardware Trojans: Adding malicious circuitry or modifying original ICs.
2. IP Piracy: Illegally pirating a design without the consent of the designer.
3. IC Overproduction: Building more than the ordered number to sell the excess in the gray market.
4. Reverse Engineering-Tampering: Tampering with ICs to manufacture counterfeits.
5. Side-channel Analysis: Extracting secret information by exploiting physical phenomena (power consumption, timing, electro-magnetic emission) involving the ICs.
6. Counterfeiting/Cloning: Illegally forging or imitating the original ICs.
7. Recycling: Reselling defective, end of service ICs.

The same works also detail on the countermeasures. The most comprehensive countermeasure is identified as the SHIELD program, by DARPA's Microsystems Technology Office [12]. A hardware named as "dielet" is at the core of the initiative. The hardware is required to be no more than a 100x100 micron circuitry, inserted in the host IC package. The circuitry has to include an encryption engine, passive sensors, near-field power and communications technologies to verify the identity and authenticity of the package, at any stage after the insertion process. By capturing tampering attempts through sensors and reporting to a secure server via radio frequency (RF) communication, the integrity of the IC package is tracked. The SHIELD's dielet is shown in Figure 1 and it is expected to have the following features [12]:

1. A circuitry that self-destructs upon any attempt to physically detach it from the host,
2. A root-of-trust for identification (ID) and cryptographic key (that never leaves the dielet) storage that it is prohibitively expensive and time-consuming to reverse engineer,
3. An on-board, complete and compact encryption engine that encrypts the communication using the stored cryptographic key,
4. Passive sensors to detect any tampering attempt on the package,
5. Inductive powering and RF communication for contactless operation,
6. Resiliency against component probing,
7. A very cheap price per unit.

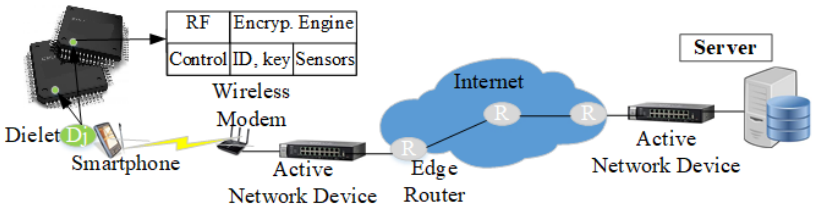

Figure 1. The 100 x 100 µm "dielet" on a host IC package. Status information of passive sensors is sent in encrypted form to a server, via the RF component

A smartphone (Figure 1) with a suitable probe is used, as the mediator between the dielet and the server [1]. The remote server contains each dielet's identification number D_j_ID, cryptographic key and the passive sensor status bits. The dielet-smartphone communication channel is a near field wireless network; while the smartphone ─ server connection is over the Internet. There are many proposals about the hardware architecture of the dielet. But, the effectiveness of the SHIELD initiative is yet to be seen, as DARPA partners with companies for IC production [14]. The recommended SHIELD authentication protocol between the dielet and the server is shown in Figure 2. The five step authentication is in fact three steps; since step 1 is merely a request and step 5 is a notification of the smartphone. Authentication protocols consisting of message exchanges less than 4 steps are not considered as mutual authentications and have known weaknesses [15]. The dielet never authenticates the smartphone or the server and message blocking is not considered. Although the channel between the smartphone and the server is assumed secure, there is nothing stopping an adversary from sending a wireless fabricated "OK" result to the smartphone, in the final step. Other works based on known attacks against computers, point to more weaknesses and vulnerabilities. For example, the sensor status bits are passed as a repeated, encrypted constant, since sensor "OK" bits are the same. The server's message freshness indicator, random number (nonce) Non_S appears both in plaintext and ciphertext form; allowing formation of a look up table. The symmetric key_{Dj} of dielet D_j is constant, instead of being variable, allowing accumulation of side-channel information [11].

B. Countermeasures Against Known Threats

Observing the disadvantages and weaknesses of the SHIELD protocol, researchers offered rectifications by proposing additions and corrections. Physically unclonable functions (PUFs) are extensively offered as a security primitive for generating the dielet Dj_ID [1], [5], [16-17]. Although PUFs can serve as a private fingerprint for each IC, they are susceptible to side-channel attacks [11]. True Random Number Generators (TRNG) found inherently in the ICs are considered natural alternatives to PUFs [1]. Using the above inherent properties, fully blown countermeasures are offered to overcome the weaknesses of the SHIELD authentication protocol. For example, claiming that Denial of Service (DoS) attack on a single dielet is possible in SHIELD, the following corrections are proposed in work [1]:

1. Adding a read-out mode before the authentication;
2. Counter mode encryption to prevent linking ciphertexts to corresponding plaintexts over time;
3. Four distinct modes: ID generation, initialization, read out and authentication modes.

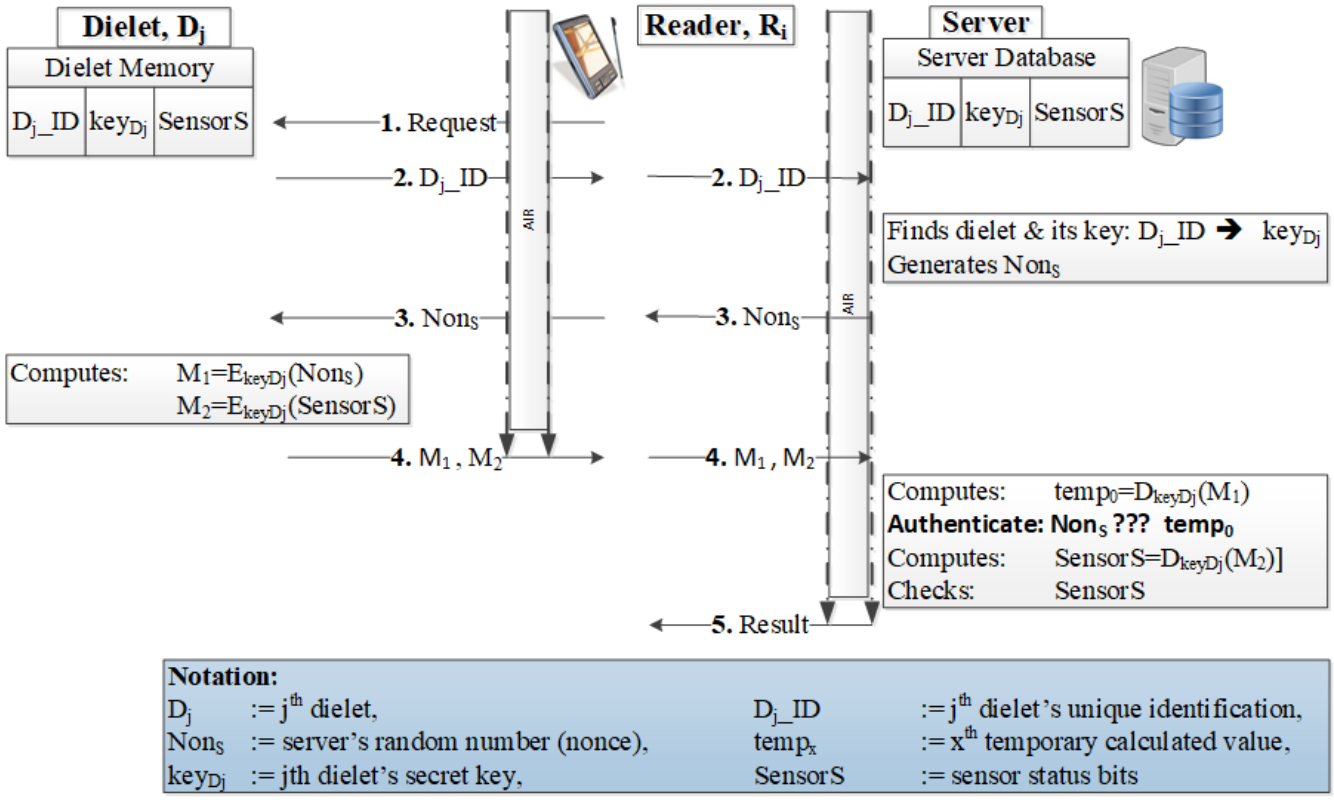

Figure 2. The SHIELD protocol, the nonce is passed both in plaintext and ciphertext form; M_2 is constant and "Result" is vulnerable.

Since work [1] is counter based, it is named as CNTR-SHIELD in present work and given in Figure 3. The number of message exchanges has been increased and the encryption has been fortified, leading the authors to claim that CNTR-SHIELD is both secure and efficient. But, the work has two surreal assumptions: The generation and initialization processes cannot be eavesdropped; the intermediary smartphone can be trusted.

However, generation phase involves wireless transmission, where malicious monitoring of electromagnetic emissions is a well-accepted threat, in security literature. Nevertheless, the encryption key_{DJ} is transmitted in plaintext to the smartphone in step 4 (Figure 3). This is an over simplistic design, putting the protocol's security in danger. Since message 4 can travel in air, beyond the smartphone; it is possible that an adversary has a chance to record the keys of the inserted dielets, with a high gain antenna. The proposal is further weakened when the dielet is recommended for "seamless integration" with Electronic Product Code tags. Tags use ultra-high radio frequency identification (RFID) technology. Numerous well known attacks on UHF tags exist in the literature, based on malicious listening. As an excuse for the weakness, the authors declare smartphone attacks outside their scope and recommend alternative security measures. However, authentication protocols are normally designed, taking the existence of rogue (dishonest, malicious, unauthorized) readers and smartphones into consideration [18-23]. Later, we will make a contribution by eliminating the threat posed by the smartphone.

Another research referring to SHIELD consists of publications by K. Yang et al. [13], [24-25]. Briefly, the works named as "RFID Enabled Supply Chain (ReSC)" and "Counterfeit Detection, Traceability, and Authentication (CDTA)" target the IoT supply chain, and stop short of offering improvement for SHIELD. Therefore, we will focus on CNTR-SHIELD and leave the analyses of ReSC and CDTA to another study. Nevertheless, the exaggerated assumptions of K. Yang et al. will be given, because the same are made in CNTR-SHIELD, as well:

1. Smartphones used for tracking are secure ─ although rogue workers exist;
2. Manufacturers and integrators are fully trusted ─ contrasting the gray market of overproduced ICs;
3. The chain of distribution is known a priori ─ contradicting unplanned/uncontrollable stops;
4. Only transitions in the supply chain are insecure ─ contradicting most sophisticated, repeated attacks launched at the start or end of the supply chain.

C. Attacks on Cntr-shield Protocol

Due to the assumptions pointed above, serious weaknesses emerge in CNTR-SHIELD. The reasons of the weaknesses and their consequences are presented below:

1. DoS Attack: Message blocking or communication loss is ignored in CNTR-SHIELD, which opens the way to desynchronization of the counter value in the dielet (C_j) and the server (C'_j). The authors downplay DoS attack, by claiming it affects a single IC package and does not require an expensive recovery process. However, this is not the case. At an intermediary location, if an adversary or a rogue employee sends the same nonce Non_{Ri} to all of the dielets in range (or one by one); the dielets increment their counters, at the authentication phase. By blocking the communication to the server, the counters at the server side are all left un-incremented, effectively desynchronizing the dielets and the server.

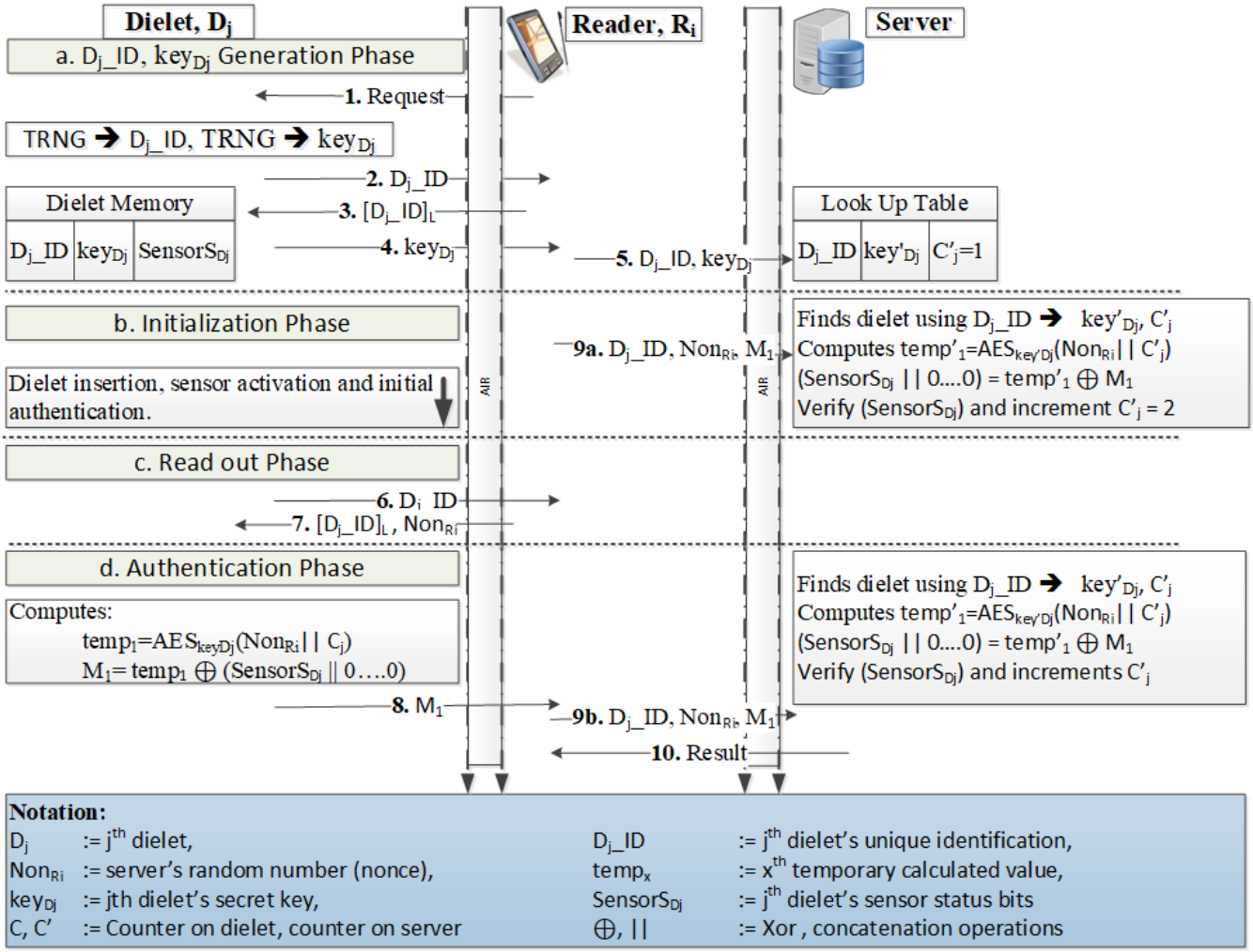

Figure 3. The CNTR-SHIELD protocol [1]

Furthermore, knowing that counter value is 2 at the exit of the manufacturer; repeating the above attack many times, until the counter reaches its maximum value, puts the dielets completely out of service. Therefore, single and bunch DoS attack is possible on CNTR-SHIELD. The same undesired result will happen, if an inexperienced employee tries to read the dielets many times, with a broken link to the server.

2. Full-disclosure Attack: In the generation phase, sending the dielet ID (D_j_ID) and the secret key_{Dj} in plaintext through air is an open breach of security principles. The manufacturer can be trusted, but malicious eavesdropping is a fact and the reason of developing security measures. Therefore, we will not try to prove the possibility of capturing plaintext messages maliciously, using a high gain antenna. The weakness makes cloning, counterfeiting and unauthorized recycling possible. For example, an adversary who captures data from generation phase, can also record the constant, encrypted sensor "OK" values, in the initialization phase of the same session. Now, the adversary has D_j_ID, key_{Dj}; knows the last hop's counter value and the sensor "OK" value. Counterfeits with foul finite state machines can be prepared, with the information in hand. At the delivery phase to the end-user (last hop), the counterfeit obtains $temp_1$ from the smartphone and prepares M_1. A server receiving the correct verification M_1 in step 9b (Figure 3) can never differentiate a counterfeit from an authentic IC.

3. Lookup Table Attack: The simple XOR operation ruins the security of the CNTR-SHIELD protocol while forming the verification message M_1. The ($SensorS_{Dj}$ || 0….0) value is constant for "OK" status. The work does not clearly define "OK" but we know that it is equivalent to "1" bits; otherwise the whole ($SensorS_{Dj}$ || 0….0) value will be zero and XORing $temp_1$ with zero would yield $temp_1$ itself. Knowing the number of sensor status bits suffices to capture $temp_1$. Simply, XORing M_1 as in Equation (1) gives $temp_1$:

$$(temp_1) = M_1 \oplus (1\ldots.1 \| 0\ldots.0) \quad (1)$$

Exposing $temp_1$ means obtaining $AESkey_{Dj}(Non_{Ri} \| C_j)$. This information puts the authors of CNTR-SHIELD face to face with their own criticism: "Ciphertexts corresponding to the same plaintext are linked over time". The time merely depends on the smartphone's different random number generation capacity. The studies investigating the generation of similar pseudo random numbers by smartphones will not be discussed in present work.

4. Replay Attack: Reception of the clipped D_j_ID from the insecure smartphone is accepted as an authentication of the server. This is a serious weakness, which allows an attacker to perform the following actions. A rogue employee at the end of the supply chain energizes a number of legitimate dielets and performs an authentication round recording the exchanged messages, without sending them to the server. By doing so, a legal set of step 9b

messages are obtained, but the dielet counters are incremented by one. Next, the rogue employee replaces equal number of legal pieces with counterfeits and replays the recorded data to the server. The counterfeits are now in service while equal number of legal ICs has been hijacked; and the counters of the hijacked pieces are now synchronized. The counterfeits are rejected in the next control. The attacker puts the hijacked, legal dielets back into service, after illegally benefiting the cost of the replacement.

III. The Proposed Mutual Authentication

The proposed method in our novel protocol involves three phases:

1. Bonding a legal user with a legal smartphone;
2. The mutual authentication of the dielet and the server, with the creation of a secure session key;
3. Dielet status tracking, by checking sensor status.

Our novel protocol is Strongly Bonded and Mutually Authenticated Protocol for SHIELD, or briefly SBMAPS. The secret keys of each dielet are enumerated and transferred to the manufacturer through a secure channel. Similarly, the IDs of the produced dielets are transferred to the IP owner, in an ordered list. The manufacturer must insert the keys into the dielets, in the ID list's order given to the IP owner. The tuples (ID, keys) are loaded into the server memory. The generation of dielet identification numbers by either using PUFs or TRNGs is preserved in our work. Therefore, dielet ID generation is omitted in our work. The assumptions of SBMAPS are more realistic than CNTR-SHIELD's and require stronger countermeasures. Our assumptions are:

1. The air channels between the dielet and smartphone; smartphone and server are both insecure;
2. Neither the smartphone nor its operator can be trusted;
3. Physically tampering the dielet and capturing the memory contents without disturbing the sensors is not possible (indispensable criterion of SHIELD);
4. Valid usr_ID, usr_key pairs for every hop are previously stored in the server database;
5. Valid smartphone cpu_IDs for every hop in the supply chain are previously enrolled to the server database.

A. Motivation and Contributions

Challenges are progressive in IC counterfeiting and have to be reciprocated by equaled counter research. Therefore, our main motivation is providing stronger security than protocols previously proposed. However, lack of support to academia for developing innovative detection and avoidance solutions is aggravating the challenge [5]. No formal announcement about an improved supply chain proves that the challenges have not yet been met, satisfactorily. The weaknesses in proposal [1] increased our motivation of providing SHIELD with better security techniques. Moreover, our proposal presented in the next section is needed not only by the defense community, but also by other consumer supply chains. Our contributions with respect to studied previous proposals are:

1. Elimination of exaggerated assumptions and the need to trusted smartphones in the supply chain;
2. Elimination of the threat by rogue employees and the need to keep sensor status bits on the dielet;
3. Strong mutual authentication of the dielet and the server at every hop in the supply chain;
4. Shared secret keys enumerated and controlled by the server;
5. New (changing) session key for every new round;
6. Optional ID update, to prevent malicious tracking.

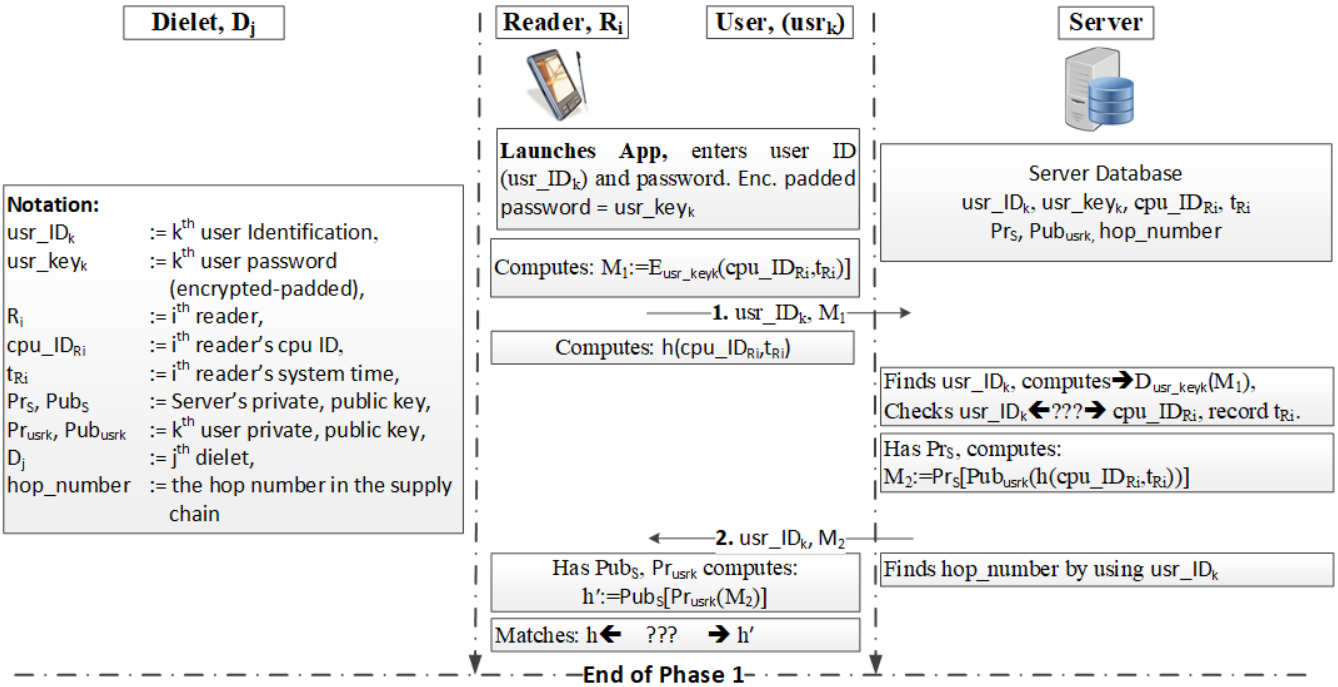

Figure 4. Phase 1 of SBMAPS "The bonding of a legal smartphone with an authenticated user"

B. Phases of Sbmaps

1. Phase 1 (User and smartphone registration and bonding): The protocol starts by bonding an authentic user with a valid smartphone, as shown in Figure 4. The dielet has no role in Phase 1. The k^{th} user has to enter his/her username usr_ID_k and password in the custom made application, on a legal smartphone. The password is padded and encrypted with a software like linux's "crypt" command to form the usr_key_k. Using the key, the application encrypts the cpu_ID_{Ri} of the smartphone together with the system time t_{Ri} to form message M_1 and sends message M_1 and usr_ID_k to the server, through insecure air medium. We will not go into the concatenation and padding before encryption, as they are encryption specific operations. Using the login name usr_ID_k the server finds the user's key usr_key_k in its database and decrypts message M_1. Server searches its database to see whether there is an entry that matches with the decrypted cpu_ID_{Ri}. After verifying that the smartphone is a valid device, the specific user is paired with it, which we define as user-smartphone bonding. The smartphone's system time t_{Ri} is recorded. Next, the server calculates the hash of the tuple (cpu_ID_{Ri}, t_{Ri}), which proves that it has the user key to decrypt message M_1. After encrypting the hash value with the user's public key and then signing the encrypted value with its private key (message M_2 is formed), the server sends message M_2 to the smartphone. While waiting, the smart phone calculates the same hash. After receiving M_2, the smartphone can decrypt M_2 and obtain server's version of the hash, if it is an authorized device; i.e. if it has the server's public and the user's private key. If the calculated and unraveled hash values match, the server has authenticated the user and bonded the user name with the smartphone, in hand. Hence, the mutual authentication of the user and server, and bonding of the user with the smartphone is finished. Meanwhile, using the user name, the server finds the hop number and decides which key number (Key_No) should be used, in the next phases of SBMAPS.

2. Phase 2 (the Mutual Authentication of the Dielet and the Server, Creation of Session Key): The bonded smartphone requests the identification of the targeted dielet, as shown in Figure 5. From then on, the smartphone is merely a repeater. The dielet sends its identification number Dj_ID to the server. The server finds the dielet in its database and determines the Key_No and the key $key\mathbf{x}_{Dj}$, by using the hop_number decided in Phase 1. After generating the random number Non_S, it is encrypted with the chosen $key\mathbf{x}_{Dj}$ key and sent to the dielet, with a prefix Key_No indicating the key used. The dielet finds the key pointed by the server and uses it to decrypt message M_3. The value $temp_0$ now contains the server's nonce. Next the dielet prepares message M_4 to show the server that it can decrypt server messages. Then, the dielet generates its nonce Non_{Dj} and hides it in message M_5. After receiving M_4 and M_5 from the dielet, the server decrypts M_4 and XORs it with M_3. The obtained value is matched with the generated Non_S. A match, authenticates the dielet. Next, M_5 is decrypted and XORed with M_4. The obtained $temp_3$ value is the nonce of the dielet. The server computes M_6 to prove the dielet that it can decrypt messages sent by the dielet. After sending M_6, the server calculates the session key key**S** by XORing the generated and received nonces. The dielet decrypts M_6 and XORs it with the previously received M_5. If the obtained value matches its generated nonce Non_{Dj}, then the server is authenticated. As a last step, the dielet calculates its own version of the session key**S**. If Phase 2 is stopped at any step, the mutual authentication has to be started from the beginning.

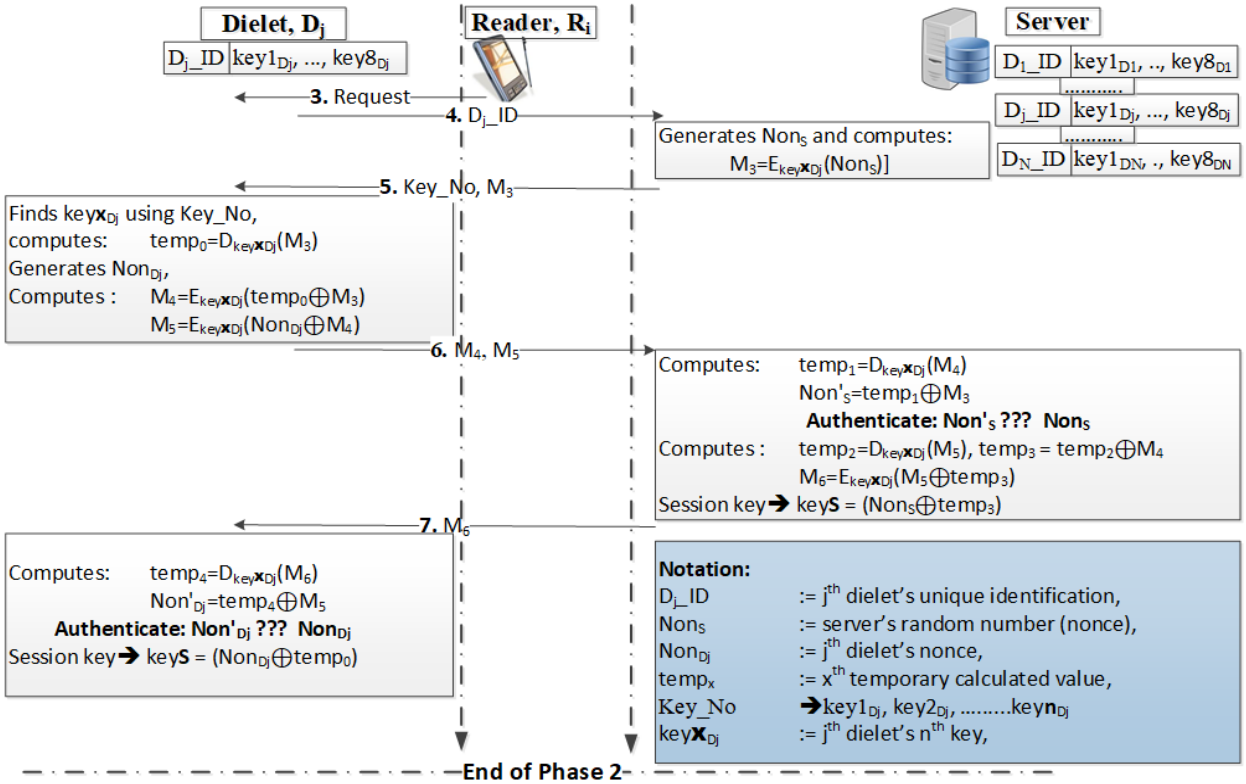

Figure 5. Phase 2 of SBMAPS "Dielet-Server mutual authentication and session key generation"

3. Phase 3 (Dielet Tracking by Sensor Status Checking): In the last phase, the server encrypts its system time with the session key and sends it to the dielet (Figure 6). The dielet decrypts the received message M_7, extracts and checks Non_{Di}. Then obtains and saves t_S to a temporary location, i.e. another register. Then, t_S is XORed with the old time old_t_S found in the IC's nonvolatile memory to form M_8. Before the dielet's first registration, initially old_t_S can be assumed to be or ANDed with zero. Next, the dielet sensors are read and their status bits are XORed with t_S and then encrypted with the session key to form M_9. Before sending messages M_8 and M_9, the new time t_S is over-written on old_t_S in IC's secure memory, dedicated for saving dielet's time of access. After receiving the messages, the server decrypts M_8 to check if the dielet knows the session key and has the correct old_t_S value. After the verification, the server decrypts M_9 to obtain the sensor status bits by XORing M_8 with $temp_8$. Then, the sensor status bits are checked. If the sensors are not reporting foul play, the time from start of Phase 1 (t_{Ri}) to the end of Phase 3 (t_{Scheck}) is checked to verify that the protocol has ended within an accepted period of time. Next, the server increments the hop_number and saves the tuple (hop_number, t_S). The server uses the accumulated tuples to track dielet's movement, in the supply chain. The server finally prepares M_{10} and sends it to the smartphone, as a notification of the result of the overall protocol. M_{10} can only be opened up by the smartphone and the result is found out by matching the calculated hashes and the received hash. At every session, the result is a variable, obscured and double encrypted value. Optionally, the server and the dielet can update (change) the D_j_ID, with the generated session key. We will not go further into the equations needed for ID update, as it will merely prolong the explanations. The dielet can update its D_j_ID, before sending messages M_8, M_9 and the server before sending M_{10}. After sending M_{10} another authentication attempt at the same hop -for the same dielet- will be dropped by the server; because a successful authentication has been completed and reported to the reader (Figure 6).

IV. Security Analysis and Comparison

For comparison equality the security and performance analysis of CNTR-SHIELD are preserved. But, the attacks demonstrated both on CNTR-SHIELD and present work will be considered in our security analysis.

A. Security Analysis of Sbmaps

Although the authors of CNTR-SHIELD present a mathematical proof of their protocol's security, four attacks have been demonstrated in Section II (C). The security analysis of SBMAPS for the same attack types is below.

1. DoS Attack: Apparently, counter desynchronization of the dielet and the server is not possible in SBMAPS, since there are no separate, synchronized counters. Message blocking, altering or loss is also not a threat, because:

a. If phases 1 or 2 are blocked, authentication fails and the protocol stops or times out;
b. If key number is altered, dielet authentication fails and the protocol halts;
c. If phases 1 or 2 are disrupted, protocol times out.

2. Full-disclosure Attack: Full disclosure is out of question in SBMAPS, as none of the messages are passed in plaintext. The ID is the only information in plaintext, which is required for tracking ICs.

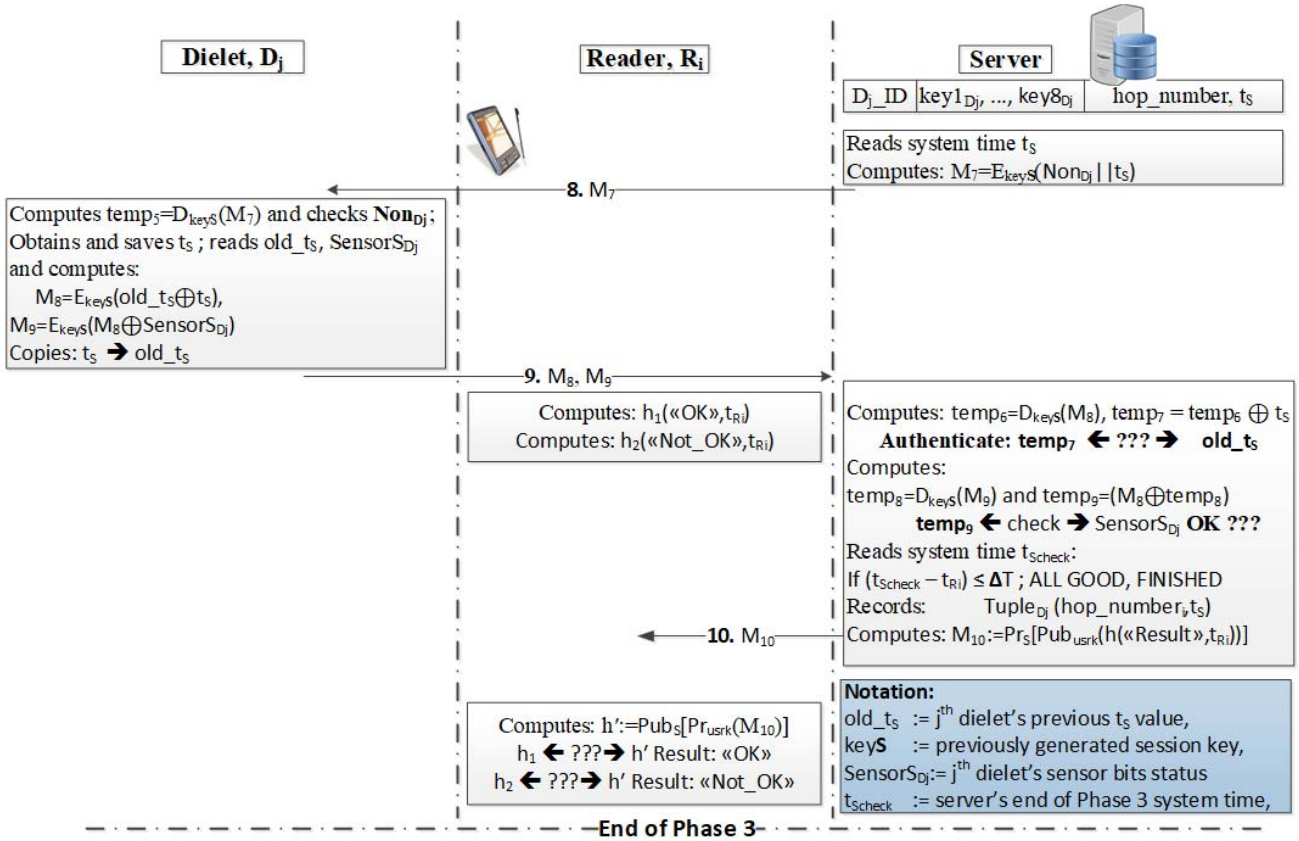

Figure 6. Phase 3 of SBMAPS "Dielet tracking by checking sensor status bits and counting hops, in the supply chain"

None of the encrypted messages exchanged can be exposed, since the shared keys and private keys are secure. Even if a rogue employee enters its credentials and hands the smartphone to an attacker, the messages cannot be decrypted; simply because the smartphone is time checked and a message relaying device, unable to see any secrets.

3. Lookup Table Attack: It is not possible to construct a plaintext-ciphertext pair table in SBMAPS, simply because no plaintext messages are passed. Additionally, the passed messages are always variable, as in the critical sensor status bits and time combination. Furthermore, messages are encrypted with a new session key, every round. The result notification of the authentication to the smartphone at the end of every round is also encrypted and variable.

4. Replay Attack: None of the message exchanged are re-playable, because the encrypted messages change every round. In addition, the authentication keys and the generated session keys used for encrypting the messages change every round.

TABLE I. SECURITY COMPARISON OF THE STUDIED THREE PROTOCOLS

Auth. Prot	TRNG	Plain Text	Sphone User	Key Update	DoS Attack	Full Disc. Attack	Lookup Table Attack	Replay Attack
SHIELD[12]	No	Yes	No	No	Yes	No	Yes	Yes
CNTR-SHIELD[1]	Yes	Yes	No	No	Yes	Yes	Yes	Yes
SBMAPS	Yes	No	Yes	Yes	No	No	No	No

B. Security Comparison of Sbmaps

Table I compares the security characteristics of the three studied protocols. CNTR-SHIELD tries to improve SHIELD protocol by generating the ID, using the dielet's inherent TRNG characteristic. There is no difference between protocols, in this property. But the dielet's ID and its secret key are sent in plaintext. In contrast, all messages are encrypted in SBMAPS. Thus, plaintext-ciphertext pairing is possible in both SHIELD and CNTR-SHIELD, the same is not true for SBMAPS. Therefore, our protocol is secure compared to the two previous schemes. The security of the intermediary smartphone and its user has critical importance. Only SBMAPS guarantees the elimination of attacks by rogue smartphones, or their users. Another important security primitive is key update. Neither of the previous protocols accommodates key update, but SBMAPS provides a new session key, every round. Hence, transmitted SBMAPS messages are changed every round, by changing the encryption keys. Due to weaknesses, the original SHIELD and CNTR-SHIELD are vulnerable to four types of attacks. On the other hand, SBMAPS does not show the same vulnerabilities and thus complies with the requirements of SHIELD. A further improvement in SBMAPS is the requirement for a login name-password tuple for each chain hop. Although many, there are a finite number of hops. Therefore, a small size password file on the server is sufficient to provide the extended security.

As a last security comparison, we will discuss the four adversarial models given in CNTR-SHIELD:

1. The attacker obtains a list of valid IDs and tries to produce a fake dielet:

As shown in Section II (C) this is possible in CNTR-SHIELD but not in SBMAPS, because the secret keys are controlled by the IP owner and all messages passed in initialization and authentication phases are encrypted.

2. The attacker has a "black box" (i.e. rogue smartphone) to access legitimate dielets and collect exchanged messages to produce a fake dielet:

As shown in Section II (C) this is possible in CNTR-SHIELD. But since all messages are variable, encrypted and controlled by the secure server, this adversarial model cannot succeed in SBMAPS.

3. The attacker has the capability to separate a dielet from its legitimate host IC and reuse the dielet in a fake IC:

Separating the dielet without setting off the passive sensors is against the requirements of SHIELD. Nevertheless, this type of attack is resisted in SBMAPS by hiding the time in non-volatile, secure IC memory. Hence, a dissected dielet cannot provide the time trace to produce a fake.

4. The attacker extracts secret keys from dielets by physical attacks to produce a fake dielet:

SBMAPS is not strong against capturing the secrets by tampering the dielet, without setting off the sensors. Physical attacks are in a different adversarial model category and have to be countered by hardware methods. Our work's scope covers the non-invasive side-channel attacks, as classified in Section II (A).

V. PERFORMANCE ANALYSIS AND COMPARISON

The present study is not challenging the hardware characteristics or ID generation methods proposed, in previous works. In present study, the security of the IC and supply chain is improved by keeping hardware equal, but designing a stronger dielet-server mutual authentication.

TABLE II. PERFORMANCE COMPARISON OF THE STUDIED THREE PROTOCOLS

Protocol Compared	# steps	# bits	# enc.	Memory (Bits)	Extra Counter	Latency
SHIELD[12]	5	448	2	178	0	2
CNTR-SHIELD[1]	10	258	1	178	1	1
SBMAPS	10	934	7	1074	0	3

Therefore, the parameters studied in the CNTR-SHIELD authentication protocol will be compared. The exchange between the smartphone and the server is not evaluated in either work, as it is a wireless communication over a high bandwidth communication through the Internet.

Table II shows the compared properties of the three protocols. The first is the number of steps taken by the protocols. SHIELD finishes verifying the dielet in 5 steps, without a mutual authentication. CNTR-SHIELD finishes verifying the dielet in 4 phases and 10 steps, also without a mutual authentication. Our SBMAPS also finishes in 10 steps, but with a strong 3 phase mutual authentication. Therefore, SBMAPS and CNTR-SHIELD have equal step size. By unrealistically declaring the smartphone as secure, CNTR-SHIELD exchanges 258 bits. SBMAPS exchanges use 934 bits, almost doubling the number used by original SHIELD. As observed, SBMAPS passes more information in equal number of steps, compared to CNTR-SHIELD. Obviously, information exchange is not efficient in CNTR-SHIELD. Similarly, our SBMAPS dielet executes a total of 7 encryption-decryption operations in order to complete one round. SHIELD and CNTR-SHIELD dielets carry out much less encryption operations, at the expense of passing plaintext data. The above three parameters impact the energy consumed by the protocols. The amount of energy consumed in message exchange is obviously higher in SBMAPS. But, dielet's energy efficiency is not a disadvantage in present context, because it is energized by the near field of the smartphone and not by a battery source. The only disadvantage of SBMAPS is the extended time taken by the dielet operations, which is a few extra micro seconds for a processor with a mega-hertz clock. While CNTR-SHIELD introduces an additional counter to overcome known ciphertext attack of constant key protocols; SBMAPS dielet instead uses 1074 bits of additional memory. Our strategy is increasing key space; hence increasing security, at the expense of extra memory. But, the extra amount of memory needed does not critically burden the die area, according to calculations made in CNTR-SHIELD, based on work [26]. Apart from the above, SBMAPS requires no additional hardware; therefore, has no critical hardware disadvantages.

While SHIELD contacts the server a couple of times, CNTR-SHIELD contacts once and loses security for that matter. SBMAPS eliminates the smartphone involvement by communicating with the server 3 times more than SHIELD, in order to mutually authenticate the dielet and the server. In other words, our protocol has roughly 3 times more latency than SHIELD, in verifying a dielet. This is a typical performance and security trade-off in computer science. This disadvantage can be diminished by increasing the communication speed between the smartphone and the server. It is obvious that our proposed SBMAPS takes longer time than previous protocols, due to strong mutual authentication. But we trust that, while making decisions security officers consider the performance/security ratio of a protocol. As such, SBMAPS is the only strong mutual authentication protocol with no weaknesses to known attacks, complying with the primary goal of SHIELD.

VI. CONCLUSION

There is evidence that counterfeit ICs on illegitimate parts have found their way into vital defense systems and other mission critical electronic devices. The illegal ICs and parts put missions, service-men into danger and cause revenue losses, in millions of dollars. DARPA has initiated the SHIELD program to encounter the penetration of foul ICs, into the supply chain. At the heart of the initiative is a root of trust called "dielet", which aims at tracking the legitimate ICs. The original authentication protocol for communicating with the dielet is not adequately strong. But, the proposed improvements also lack security against side-channel attacks, which are launched by eavesdropping on the exchanged dielet-server messages. Our proposed SBMAPS scheme identifies and bonds a legal user with a legal smartphone. Even if the credentials of the user are stolen, the proposed protocol degrades the role of the intermediary smartphone into a message repeater, between the dielet and the server. No nonces, secrets or messages are passed in plaintext. Thus exposing the secrets of the dielets for creating fakes is rejected. Comparison shows that the energy efficiency of our proposed SBMAPS is poorer and communication latency is higher than previous works. But, our proposed protocol is secure against four types of attacks; while previous works are vulnerable. The decision reduces to the common dilemma of preference between performance and security. Our evaluation is that the essence of SHIELD initiative dictates preferring security. The only requirement of our proposed protocol is increasing the smartphone-server communication speed.

REFERENCES

[1] C. Jin, M. van Dijk, "Secure and efficient initialization and authentication protocols for SHIELD," IEEE Transactions on Dependable and Secure Computing, vol. 16, p. 156–173, 2019. doi:10.1109/TDSC.2017.2647950

[2] U.S. Defense Science Board Task Force on High Performance Microchip Supply, "Defense science board task force on high performance microchip supply," Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, 2005.

[3] G. Sumathi, L. Srivani, D. T. Murthy, A. Kumar, K. Madhusoodanan, "Hardware obfuscation using different obfuscation cell structures for PLDs," in A Systems Approach to Cyber Security: Proceedings of the 2nd Singapore Cyber-Security R&D Conference (SG-CRC 2017), Singapore, Feb. 21-22, 2017, pp. 143-157. doi:10.3233/978-1-61499-744-3-143

[4] S. Leef, "Supply chain hardware integrity for electronics defense (SHIELD)," Defense Advanced Research Projects Agency (DARPA) Microsystems Technology Office, 2018.

[5] U. Guin, K. Huang, D. DiMase, M. Tehranipoor, Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," in Proceedings of the IEEE, vol. 102, no. 8, pp. 1207–1228, Aug. 2014. doi:10.1109/JPROC.2014.2332291

[6] U.S. Department of Commerce Bureau of Industry and Security Office of Technology Evaluation, "Defense industrial base assessment: counterfeit electronics," Jan. 2010.

[7] U.S. Senate Committee on Armed Services 112th Congress, "Inquiry into counterfeit electronic parts in the department of defense supply chain," May 2012.

[8] S. Skorobogatov, "Hardware assurance and its importance to national security," May 2012.

[9] M. J. Blair, "Proactive defense for evolving supply chain counterfeiting," Air Command and Staff College, Air University, Maxwell AFB, USA, Rep. AU/ACSC/2015, Dec. 2015.

[10] S. Leef, "Supply chain hardware integrity for electronics defense," Defense Advanced Research Projects Agency (DARPA) Microsystems Technology Office, 2019.

[11] M. Rostami, K. Farinaz, R. Karri, "A primer on hardware security: Models, methods, and metrics," in Proceedings of the IEEE, vol. 102, no. 8, pp. 1283–1295, Aug. 2014. doi:10.1109/JPROC.2014.2335155

[12] Defense Advanced Research Projects Agency (DARPA), "Tiny, cheap, foolproof: seeking new component to counter counterfeit electronics," Microsystems Technology Office (MTO) Broad Agency Announcement, 2014.

[13] K. Yang, D. Forte, M.M. Tehranipoor, "CDTA: A comprehensive solution for counterfeit detection, traceability, and authentication in the IoT supply chain," ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 22, no. 3, May 2017. doi:10.1145/3005346

[14] F. Koushanfar, R. Karri, "Can the SHIELD protect our integrated circuits?," IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS), Aug. 2014. doi:10.1109/MWSCAS.2014.6908424

[15] G. Dalkilic, M.H. Özcanhan, H.Ş. Çakir, "Increasing key space at little extra cost in RFID authentications," Turkish Journal of Electrical Engineering & Computer Sciences, Dec. 2013. doi:10.3906/elk-1201-77

[16] M. Rostami, M. Majzoobi, F. Koushanfar, D.S. Wallach, S. Devadas, "Robust and reverse-engineering resilient PUF authentication and key-exchange by substring matching," IEEE Transactions on Emerging Topics in Computing, vol. 2, no. 1, pp. 37–49, Mar. 2014. doi:10.1109/TETC.2014.2300635

[17] U. Rührmair, D. Srinivas, F. Koushanfar, "Security based on physical unclonability and disorder," Introduction to Hardware Security and Trust, pp. 65-102, USA:Springer, 2012.

[18] M.H. Özcanhan, G. Dalkılıç, S. Utku, "Cryptographically supported NFC tags in medication for better inpatient safety," Journal of Medical Systems, vol. 38, no. 8, pp. 1-15, Aug. 2014. doi:10.1007/s10916-014-0061-x

[19] Y.C. Yen, N.W. Lo, T.C. Wu, "Two RFID-based solutions for secure inpatient medication administration," Journal of Medical Systems, vol. 36, no. 5, pp. 2769–2778, Oct. 2012. doi:10.1007/s10916-011-9753-7

[20] P. Peris-Lopez, A. Orfila, A. Mitrokotsa, J.C.A. Van der Lubbe, "A comprehensive RFID solution to enhance inpatient medication safety," Int. Journal of Medical Informatics, vol. 80, no. 1, pp. 13–24, Jan. 2011. doi:10.1016/j.ijmedinf.2010.10.008

[21] M.H. Özcanhan, S. Baytar, S. Utku, G. Dalkılıç, "security issues of a recent rfid multi tagging protocol," International Journal of Advanced Computer Science and Applications(IJACSA), vol. 6, no. 1, pp. 11–15, Jan. 2015. doi:10.14569/IJACSA.2015.060102

[22] P. Peris-Lopez, A. Orfila, J. C. H. Castro, J. C. A. Lubbe, "Flaws on RFID grouping-proofs. Guidelines for future sound protocols," Journal of Network and Computer Applications", vol. 34, No. 3, pp. 833–845, May 2011. doi:10.1016/j.jnca.2010.04.008

[23] M.H. Özcanhan, "Improvement of a weak RFID authentication protocol making drug administration insecure," Life Science Journal, vol. 11, no. 10, pp. 269–276, Jan. 2014.

[24] K. Yang, D. Forte, M.M. Tehranipoor, "Protecting endpoint devices in IoT supply chain," presented at the 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, USA, Nov. 2-6, 2015. doi:10.1109/ICCAD.2015.7372591

[25] K. Yang, D. Forte, M.M. Tehranipoor, "ReSC: An RFID-enabled solution for defending IoT supply chain," ACM Trans. Design Automation of Electronic Systems, vol. 23, no. 3, Apr. 2018. doi:10.1145/3174850

[26] D.S. Jeong, R. Thomas, R. Katiyar, J. Scott, H. Kohlstedt, A. Petraru, C. S. Hwang, "Emerging memories: resistive switching mechanisms and current status," Reports on Progress in Physics, vol. 75, no. 7. doi:10.1088/0034-4885/75/7/076502