Secure Goods Supply Chain and Key Exchange with Virtual Proof of Reality

Yansong Gao[1,2], Damith C. Ranasinghe[2], Said F. Al-Sarawi[1], and Derek Abbott[1]

[1] School of Electrical and Electronic Engineering,
The University of Adelaide, SA 5005, Australia,
{yansong.gao, said.alsarawi, derek.abbott}@adelaide.edu.au,
[2] Auto-ID Labs, School of Computer Science,
The University of Adelaide, SA 5005, Australia,
damith.ranasinghe@adelaide.edu.au

Abstract. A new security protocol of *virtual proof of reality* (VP) is recently proposed by Ruhrmair *et al.* The VP allows one party, the prover, making a physical statement to the other party, the verifier, over a digital communication channel without using any secret keys except the message sent between these two parties. The physical statement could be a physical feature—eg. temperature—or phenomena—eg. destruction—of the hardware in the prover's system. We present two applications—secure key exchange and secure goods supply chain—building on the VP of temperature, location, and destruction. Moreover, we experimentally demonstrate the first electrical circuit-based VP of destruction through the proposed hardware security primitive—a hybrid memristor and physical unclonable function (memristor-PUF) architecture, which takes advantage of the PUF extracted from static variations of CMOS devices inherent to the fabrication process and dynamic variations attributed to switching variabilities of nano memristors.

Keywords: virtual proof of reality, physical unclonable function, PUF, hardware security, memristor, model building attacks, supply chain, key exchange, authentication.

1 Introduction

A new security protocol of *virtual proof of reality*, or simply *virtual proof* (VP) is recently proposed by Ruhrmair *et al* [1] in IEEE Symoposium on Security and Privacy, 2015. The VP enables one party (eg. prover) situated in an untrusted environment to prove a physical statement of a witness object (WO) over an insecure communication channel to the other party (verifier), where the physical statement could be temperature, position, voltage and the WO could be a physical unclonable function (PUF) sensitive to temperature/position or other physical object, eg. quantum systems. The difference between VP and traditional security protocols is that the WO does not store secret keys. Ruhrmair *et al* [1] demonstrate the VP of temperature, position, and modification/destruction of the WO through experiments.

The VP relate to, and extend several known concepts in cryptography and security. For example, VP extends the work of sensor physical unclonable functions (sensor PUFs)[2] to VP of temperature and position, and further generalize the VP of other sensor data. The VP is an independent and complementary protocol of physical zero-knowledge protocols proposed by Fisch *et al* [3]. One of the difference is that Fisch *et al* [3] deal with the theory, without giving implementations. The other difference is that VP and physical zero-knowledge protocols assume different adversarial models, where, in [3], the verifier and the prover may be in the same place, each possessing their own, trusted and unmanipulated detector or measurement device, while VP assumes that the verifier and the prover locate in different places, and both do not have trusted sensors and detectors.

In this paper, our contributions are threefold: i) we extend the VP of temperature and position to secure key exchange as an alternative to resorting to public key cryptography to securely transfer a secret key between parties. ii) we demonstrate the first electrical circuit-based construction of VP of modification/destruction of an object based on a new hardware security primitive combining memristors and the PUF (memristor-PUF), which utilizes the dynamic randomness of memristor inherent in its reprogramming [4] and the static random responses of the PUF induced from the uncontrollable fabrication process [5, 6]. iii) We show that the VP of modification/destruction of the memristor-PUF is able to secure transmission of goods even the supply chain is untrusted, furthermore, it can be used to secure key exchange as well with faster speed.

Organization of this paper: Section 2 presents work related to this paper. Section 3 extends the VP of temperature and location to achieve secure exchange of private key for symmetric key cryptography without the use of public key cryptography—as is the convention. The first electrical circuit-based construction of VP of destruction is demonstrated in Section 4. Furthermore, its application on securing goods supply chain and fasting key exchange are presented and discussed in Section 5. The last Section 6 concludes this paper.

2 Related Work

2.1 VP of Reality

It is assumed two parties located in spatially separated locations, where the prover is with system S_1 and the verifier is with system S_2 communicating with each other through a digital communication channel as shown in Fig. 1. The prover wants to prove a physical statement to the verifier. The physical statement is a physical feature such as temperature or phenomenon such as destruction of the system S_1 that integrates with a WO. On one hand, the physical statement made by the prover should convince the verifier with a high probability that it is true. For example, the prover convince that the verifier the temperature of

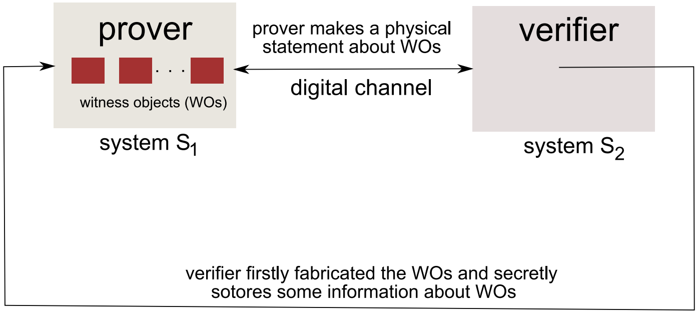
prover
witness objects (WOs)
system S_1
prover makes a physical statement about WOs
digital channel
verifier
system S_2
verifier firstly fabricated the WOs and secretly sotores some information about WOs

Fig. 1. The general setting of VP based on witness objects (WOs). The WOs shall neither contain secret keys nor be assumed as tamper-resistant. The prover makes a claim such as a physical feature or phenomena about WOs.

system S_1 is within a specific range. On the other hand, the verifier can also realize that the physical statement that the prover attempts to make a false statement with high probability if the physical statement does not true.

The VP can be classified into the private VP and the public VP according to the requirement that which party is responsible for fabricating the WO. As for private VP, the WO is only allowed to be prepared and fabricated by the verifier. The verifier measures some information of the WO and stores it secretly. Then the WO is shipped to the prover before the start of the actual virtual proof of physical statement, see Fig. 1. As for the public VP, the WO is allowed to be fabricated by either party. There is no need to transfer the WO among different parties. Moreover, the measured information on the WO is stored publicly to any parties. The public VP is possible by the use of public PUF or SIMPL systems [7, 8]. We would like to note that in this paper, we only consider the private VP since the public VP seems facing challenge in practice due to the reliability of the public PUF or SIMPL systems [7]. Therefore, the VP in this paper actually refers to private VP. However, the applications proposed in this paper can also be achieved through deploying public VP.

The VP is built upon a suitable witness object (WO). The WO shall not contain any classical secret keys (eg. digital strings stored in non-volatile memory) nor be assumed actively tamper-resistant [1]. Ruhrmair *et al.* exploit the PUF to experimentally demonstrate VP of temperature, location, and destruction. As a consequence, following in this section, we firstly introduce PUF as the deployed WO to demonstrate the key exchange application built upon VP, specifically, VP of temperature and position. Next, we introduce memristor as we are going to exploit special characteristics of it to demonstrate the first electrical circuit-based VP of destruction.

2.2 Physical Unclonable Function

The PUF is a novel hardware security primitive, acts as a 'fingerprint' of a hardware device. The PUF, especially silicon based PUF [9, 6], offers a simple alternative to storing digital keys in NVM (non-volatile memory) with a small hardware footprint and without the need for tamper-sensing mechanisms for extracting secret key information from a complex physical system. Notably that PUFs are easy to build but practically impossible to duplicate because they rely on uncontrollable physical parameter variations that occur during hardware device manufacturing. As a fact that the secrecy of a PUF is derived from the inherent complicated physical system instead of storing information in NVM memories, the PUF is inherent resistant to invasive tempering, eg. probing and depackaging. These invasive attacks will inevitably destroy the PUF itself with high probability. Therefore, no useful information is able to be obtained through such attacks.

Different PUFs with the same design and same fabrication process result in different responses (output) when the same challenge (input) stimulated to the PUFs. While the same PUF results similar responses when the same challenge applied to it at different times and under different ambient environments. All of these two features ensure the verifier could identify a special PUF in a large population according to its unique challenge response pairs (CRPs) derived from the randomness when it is born. Over the years, a number of PUF structures have been proposed, built and analyzed. These include *time delay based* PUFs such as the Arbiter PUF [9–11] (APUF), Feed-Forward APUF [5], Ring-Oscillator PUF [6] (RO-PUF), and Glitch PUF [12]; *Memory-based* PUFs leveraging device mismatch such as SRAM PUF [13, 14], Latch PUF [15], Flip-flop PUF [16, 17], Butterfly PUF [18]. A comprehensive review of different PUF architectures can be found in [19, 20]. In recent years, emerging PUFs with nanotechnology are initially investigated aiming to build PUFs beyond the aforementioned conventional silicon PUFs by taking advantage of prevalent process variations as a consequence of scaling down to the nano region, and other unique properties offered in emerging nanoelectronics devices [21–24]. A review of such nano PUFs can be found in [25].

The instability of PUF is undesirable and needs to be minimized. However, the VP of temperature actually takes this undesirable property to an advantage. In other words, the responses is dependent not only on the response itself but also the temperature the PUF is in. Therefore, the CRP can be used to identify the temperature of the PUF that works in. This is similar to the sensor PUF [2]. Where the sensor PUF makes the sensor and crypto module inseparable by merging sensing with cryptography to ensure the authenticity and veracity of measurements of a physical quantity (PQ)—the value of the temperature, voltage, and distance—in an untrusted remote environment.

2.3 Memristor

A memristor is a two terminal non-volatile nano memory element. Its resistance can be switched between high resistance state (HRS) and low resistance state (LRS), where the HRS and LRS are two logic states for storing digital information. The resistance of memristor can also be tuned to any intermediate value between the HRS and the LRS according to the width and amplitude of the programming pulse across it [4]. The non-volatility relies on the fact that the resistance of the memristor remains unchanged when the power is off. The memristor is considered as a promising NVM due to its smaller footprint, faster switching speed, higher endurance, lower power consumption and longer retention time. Unfortunately, memristors suffer significant performance variability attributed to physical nanoscale variations. The resistance variation is not only from variations in geometry—eg. thickness, doping—determined by uncontrollable fabrication process variations but also from C2C variation due to the random locations of some filaments in the memristor—these metal filaments are formed and disrupted during HRS/LRS reprogramming.

In section 4, we propose to utilize the C2C variation as an advantage to facilitate building a highly secure hybrid hardware security primitive—memristor-PUF. This memristor-PUF is utilized to demonstrate the first electrical circuit-based VP of modification/destruction.

3 Key Exchange Build Upon VP of Temperature and Location

Cryptographic applications require secure key exchange between parties before a secure communication channel is set up. Public key cryptography is usually used to solve the private shared key distribution problem. In this section, we rely purely on the PUF as an alternative to resorting to public key cryptography to securely transfer a secret key between parties. Ruhrmair *et al.* use the PUF to experimentally demonstrate the VP of temperature and location. While we extend it to secure key exchange.

3.1 Preparation of WO

The foundation of VP is a WO that is prepared by the verifier and handed over to the prover through an untrusted supply chain prior to the VP application. To achieve secure key exchange based on VP, firstly, a WO is needed. In this paper, we employ a strong PUF [26, 25] as a necessary WO as shown in Fig. 2. The strong PUF acting as the WO will be transferred from the verifier to the prover. Secondly, the strong PUF acting as a WO must satisfy the following security properties:

Property 1. The strong PUF is assumed to be PQ dependent in its behaviour. Specifically, the response $\mathbf{R_j^i}$ of the strong PUF is not only a function of the challenge $\mathbf{C_j}$, but also a function of its PQ$_i$. In other words, $\mathbf{R_j^i} = F_{\mathrm{PUF}}(\mathbf{C_j}, \mathrm{PQ}_i)$. A very different $\mathbf{R_j^i}$ is desirable for the same $\mathbf{C_j}$ under different discretized PQ$_i$.

Property 2. The $\mathbf{R_j^i}$ is insensitive to other variations except the specific PQ. For example, if the specific PQ is temperature, then a PUF's response should be stable against voltage variations.

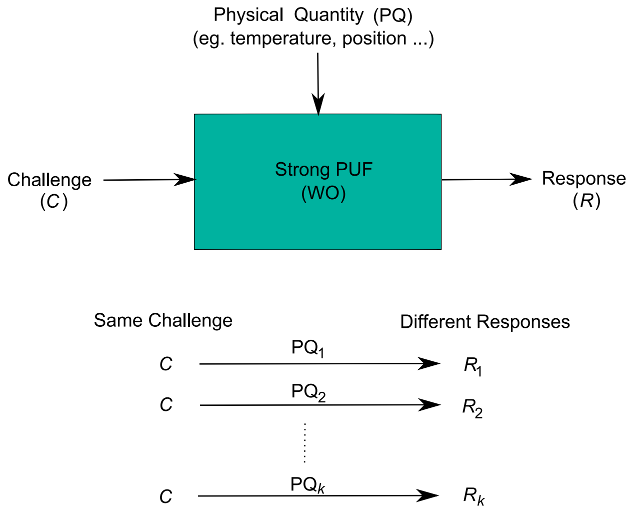

Fig. 2. Response (R) of the PUF is a function of the Challenge (C) and the Physical Quantity (PQ). For the same C, different R are produced due to the difference of PQ.

Property 3. The PUF is resistant to model building attacks, which implies that knowing many $\mathbf{R^i_j} = F_{\mathrm{PUF}}(\mathbf{C_j}, \mathrm{PQ}_i)$ for various $\mathbf{C_j}$ and PQ_i, an adversary can not predict the unmeasured $\mathbf{R^s_r}$ for new $\mathbf{C_r} \neq \mathbf{C_j}$ or new $\mathrm{PQ}_s \neq \mathrm{PQ}_i$. This is actually a property of a strong PUF.

3.2 Protocol

The protocol is split into two phases: the enrolment phase and the key exchange phase.

Enrolment Phase

1. The verifier prepares a strong PUF_A acting as the WO that is dependent on a specific PQ.
2.
```
 for $i = 1 : k$
 set $\mathrm{PUF}_A$ under $\mathrm{PQ}_i$;
  for $j = 1 : m$
   randomly select $\mathbf{C^i_j}$, apply $\mathbf{C^i_j}$ to $\mathrm{PUF}_A$ and measure $\mathbf{R^i_j}$;
  end
end
```
3. The CRP database (DB) is created and saved. Here DB $= \{\mathbf{C^i_j}, \mathbf{R^i_j}, \mathrm{PQ}_i\}$ for $i = 1, ..., k$ and $j = 1, ..., m$.
4. Then the WO of PUF_A is transferred to the prover.

Key Exchange Phase The key exchange protocol follows the listed steps and is illustrated in Fig. 3:

1. The verifier randomly selects a $\mathbf{C_j}$ from its DB of PUF$_A$ acting as the WO and sends it to the prover.
2. According to the key that needs to be exchanged, the prover sets PUF$_A$ under a specific PQ$_i$ and applies $\mathbf{C_j}$ to PUF$_A$ to obtain the $\mathbf{R^i_j}$. Hence, $\mathbf{R^i_j}$ contains the information of PQ$_i$ where the response $\mathbf{R^i_j}$ is transmitted without encryption. For example, if PQ is temperature, then $T_1, T_2, ..., T_k$, where $k = 8$, can be encoded as 000, 001, 010, 011, 100, 101, 110, 111. If a key of 010 needs to be transferred, then PUF$_A$ is measured under PQ $= T_3$. Hence, $\mathbf{R^3_j}$ is acquired by the prover.
3. The verifier receives the $\mathbf{R^i_j}$ and compares all stored $\mathbf{R'^i_j}$ in its DB with $\mathbf{R^i_j}$. If one of $\mathbf{R'^i_j}$ matches $\mathbf{R^i_j}$, then the key encoded by PQ$_i$ is accepted. Otherwise, if none of $\mathbf{R'^i_j}$ matches $\mathbf{R^i_j}$, this key exchange round is aborted. For example, the verifier receives $\mathbf{R^3_j}$, then compares it with $\mathbf{R'^i_j}$, where $i = 1, 2, ..., 8$, with $\mathbf{R^3_j}$. Only the $\mathbf{R'^3_j}$ will match $\mathbf{R^3_j}$. Then the encoded key bits 010 is discovered by the verifier.
4. If the transferred key is long, then the key will be split into short length partial keys. Steps 1–3 will be repeated until the entire key is completely transferred.

Note that the PQ$_i$ of the WO is encoded as different digital values while the encoding scheme can be public known. In our key exchange protocol, once the verifier successfully discovers the PQ$_i$ of the WO, the corresponding key bits can be fully determined.

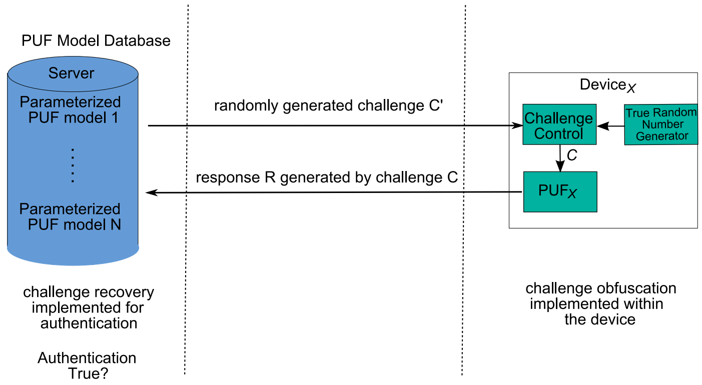
PUF Model Database
Server
Parameterized PUF model 1
Parameterized PUF model N
randomly generated challenge C'
response R generated by challenge C
Device$_X$
Challenge Control
True Random Number Generator
C
PUF$_X$
challenge recovery implemented for authentication
Authentication True?
challenge obfuscation implemented within the device

Fig. 3. Key exchange protocol.

3.3 Discussion and Proofs

Why is this key exchange protocol secure? In our scheme, security is achieved by the adversary's inability to obtain the key either through eavesdropping or from prediction by building a model of the strong PUF acting as the WO. In contrast, the verifier can successfully discover a specific PQ_i selected by the prover and then the corresponding key. We can ensure that only the verifier can discover the key because of the security properties ensured by the strong PUF used as the WO:

1. According to **Property 3**, although the adversary may physically access PUF_A, it is infeasible to measure all of the CRPs within a period (eg. several days or months) due to the significant population of CRPs generated from a strong PUF. Moreover, the adversary cannot predict $\mathbf{R_r^s}$ for unused $\mathbf{C_r}$ or PQ_s.
   Therefore, the **Property 3** of PUF_A used as a WO ensures that the adversary cannot impersonate the PUF_A through a mathematical model or create a CRP database consisting of all the CRPs generated from PUF_A. Hence, it ensures that the prover is the authentic party that the verifier communicates with. Because only the prover who holds the PUF_A can obtain any $\mathbf{R_j^i}$ under a specific PQ_i corresponding to a randomly selected $\mathbf{R_j}$ sent from the verifier.
2. According to **Property 2** and as illustrated in Fig. 4, the Fractional Hamming Distance among all $\mathbf{R_j^i}$ measured under different PQ_i $(i = 1, 2, ..., k)$ to the same challenge $\mathbf{C_j}$ named as PQ-FHD is always larger than bit error rate (BER) due to other variations. Where the BER is the FHD among $\mathbf{R}$ corresponding to the same $\mathbf{C}$ and PQ but for the $\mathbf{R}$ evaluated multiple times. In general, BER is a consequence of measurement noise or other environmental variations.
   Therefore, the **Property 3** of PUF_A used as a WO allows only the verifier the ability to find the PQ_i in order to discover the key encoded by the prover. This strategy originates from [27].

We now demonstrate that the necessary security properties of a WO can be met by two typical strong PUFs using experimental data in [1] and therefore show the practicability of our proposed protocol. Specifically, temperature and position are used respectively to show that different PQs of the WO can be utilized to satisfy our key exchange protocol in practice. To verify that temperature can be used as a PQ, a 4 XOR-Bistable Ring PUF [28] (4 XOR BR-PUF) is tested. To verify that position can be treated as a PQ, an optical PUF [29] is used and tested. Note both of 4 XOR BR-PUR and optical PUF are strong PUFs.

The performance of the two metrics that are considered for our proposed key exchange protocol are shown in Table 1. As we can see, average PQ-FHDs are always higher than the BER for different discretized PQs. Especially when the PQ of position is used. In terms of the performance of the 4 XOR BR-PUF, the reason for the difference between PQ-FHD and BER not being large is that the difference between PQ_i and PQ_{i+1} is 4 °C. If the temperature difference

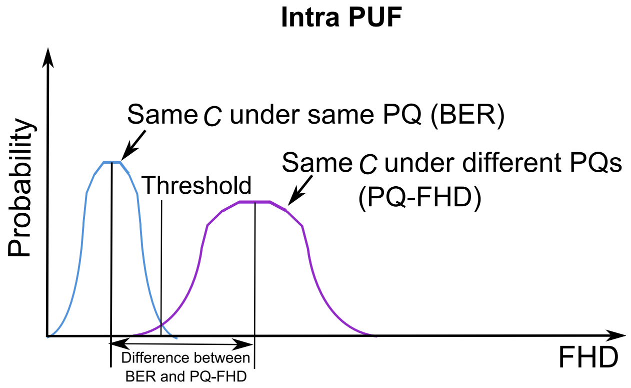

Fig. 4. Fractional Hamming Distance (FHD) Distribution. Evaluation is carried out for the same PUF. The term ‘Intra PUF’ indicates that evaluations on both the BER and PQ-FHD performance refer to the same PUF instance.

between PQ_i and PQ_{i+1} increases to 8 °C, the PQ-FHD will be higher than 6.2%, while the BER stays unchanged and consequently further increase the probability of successfully recovering the encoded key corresponding to PQ_i. In addition, the PQ-FHD performance can be further improved by increasing the PUF’s sensitivity to temperature, which provides scope for future work.

Therefore we can see that the large FHD difference between the BER and the PQ-FHD provides the verifier with the ability to successfully recover the encoded key. However, the adversary is unable to recover the key except through brute force.

We would like to note the experimentally proof of secure key exchange protocol based on VP of temperature and location might not satisfy with requirements if the key exchange speed is fast. However, it is observed the response is more sensitive to voltage compared with temperature [30]. Therefore, voltage is expected to be the PQ to speed the key exchange fast.

Table 1. Average BER and PQ-FHD performance under different PQs

PQ	BER	PQ-FHD	WO
Temperature	1.0%	6.2%	4 XOR BR-PUF
Position	8.7%	36.3%	Optical PUF

4 VP of Destruction

In this section, we first introduce the definition of the VP of destruction. Secondly, we propose a hybrid new security primitive of memristor-PUF. As a consequence, we demonstrate that the memristor-PUF can be utilized to satisfy the requirements of VP of destruction. To the best of our knowledge, this is the first electrical circuit-based construction of VP of destruction.

4.1 Definition

The VP of destruction is that the prover wants to prove that a certain object in the prover's system S_1 is irreversibly destroyed, or modified, to the verifier.

As suggested in [1], the VP of destruction is possible to be implemented as the following manner:

1. Firstly, the prover shows that the object OB_1 is in his possession.
2. Secondly, the prover destroys or modifies the OB_1 to obtain the second object OB_2. To get OB_2, the prover must irreversible destroy or modify the OB_1.
3. Thirdly, the prover proves that the OB_2 is also in his/her possession.

In [1], the authors experimentally demonstrate two constructions to prove this security concept, VP of destruction. One construction relies on optical mechanical and the other one relies on quantum mechanical. Both of them are different from our proposed construction that relies on a simple electrical circuit.

4.2 Memristor-PUF Design

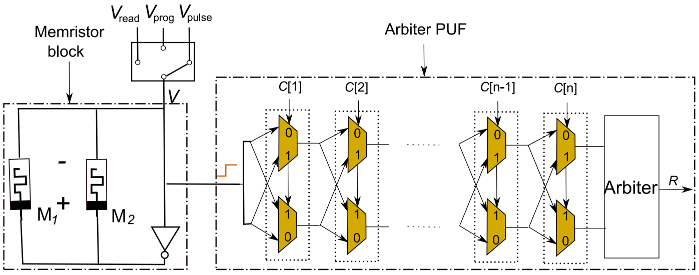

Fig. 5. Schematic of the memristor-PUF. The APUF (Arbiter PUF) is chosen for demonstration. while other PUF structures can also be employed.

Arbiter PUF The schematic of a hybrid memristor and PUF security primitive—memristor-PUF is shown in Fig 5. We deploy the APUF (Arbiter PUF) for demonstration due to its simplicity and capability in generating an exponential number of CRPs [31]. However, The APUF is vulnerable to modeling attacks [26]. Note that other types of PUFs can also be employed to substitute the APUF with appropriate circuit modification. The Arbiter PUF is responsible for producing CRPs. The APUF consists of n stages in sequence, each stage is composed of two 2-input multiplexers shown in Fig. 5, or any other architectures that have two signal paths. To generate a response bit, a signal $V = V_{\text{pulse}}$ is stimulated to the first stage input, while the challenge $\mathbf{C}_i$ is used to select the signal path in each chain to the next stage. These two electrical signals simultaneously race through each multiplexer path (top and bottom paths) in parallel. At the end of the APUF architecture, an arbiter, which can be implemented by a latch, is used to determine whether the top or bottom signal arrives first and hence outputs a logic ‘0’ or ‘1’ accordingly.

Experimental Validation of C2C Variation of Memristors The memristor switches from HRS to LRS with a negative potential difference between the bottom electrode and top electrode, marked as ‘+’ in Fig. 5, corresponding to SET switching as one or more conductive filaments grow or form, while it switches from LRS to HRS with a positive potential difference between the bottom electrode and top electrode corresponding to RESET switching as the filaments are disrupted—as shown in Fig. 6a. Note this positive potential enabled RESET operation is the case for the device we investigated and in other devices they can be SET by a positive voltage. The memristor has unique C2C variation induced by each programming operation. This phenomenon is caused by the random change of locations of some filaments during disruption and formation.

To validate the obvious C2C variation in HRS, we fabricated a number of memristors and tested them. A 50 nm thin film of $SrTiO_3$ is deposited on a $Pt/Ti/SiO_2$ (50:10:300 nm) pre-patterned Si substrate using RF magnetron sputtering at room temperature, from a stoichiometric ceramic target. Top Pt/Ti (50:10 nm) electrodes are fabricated by three-step photolithography/lift-off processes and deposited by using electron beam evaporation at room temperature. A detailed description of fabrication is in [**?**,**?**]. The photomicrograph of fabricated memristors are shown in Fig. 7. The characterization of devices was performed by pulse transient measurements using a sourcemeter (Agilent 2912A). From our measurements, the obvious C2C variation is observed as shown in Fig. 8. Hence, the resistance observed on HRS state or HLS state varies from cycle to cycle, see Fig. 8. This phenomenon has been used to build reconfigurable PUFs [23, 32].

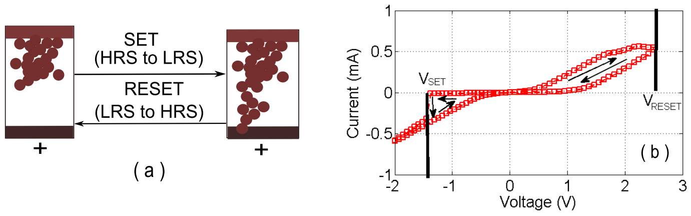

Fig. 6. (a) Illustrates the SET/RESET operation principles of a memristor. (b) The current-voltage characteristic of a memristor we fabricated. The $V_{\text{SET}} = -1.5$ V and $V_{\text{RESET}} = 2.5$ V are observed. (d) Photomicrograph showing our array of fabricated memristors on a die. (e) Photomicrograph of single memristor device.

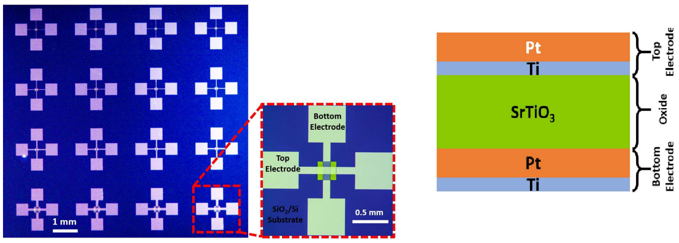

Fig. 7. Photomicrograph showing our fabricated memristors on a die.

Table 2. Voltage value settings to meet with our proposed security of the memristor-PUF

V_{read}	$> V_{\text{SET}}$ && $< V_{\text{RESET}}$
V_{prog}	$< V_{\text{SET}}$ || $> V_{\text{RESET}}$
V_{pulse}	$> \lvert V_{\text{SET}}\rvert > \lvert V_{\text{read}}\rvert$

Operation Principles The memristor is also observed having threshold voltages V_{SET}, (negative) and V_{RESET} (positive), see Fig. 7c. The resistance change is slow or negligible when $V_{\text{SET}} < |V| < V_{\text{RESET}}$. Therefore, the read operation is carried out by applying a small voltage $V = V_{\text{read}}$, where $V_{\text{SET}} < V_{\text{read}} < V_{\text{RESET}}$, eg. 300mV. Conversely, the memristor switches abruptly when $V > V_{\text{RESET}}$ or $V < V_{\text{SET}}$. The voltage settings are listed in Table 2.

To activate path race inside APUF, the V will be connected to $V_{\text{pulse}} > |V_{\text{SET}}| > |V_{\text{read}}|$, see Fig. 5, to enable the response generation. Otherwise, responses are unable to obtained. It is clear that CRP evaluation will irreversibly switch memristors to LRS state considering that they are pre-programmed to be in HRS. The memristors permanently record, counting on the non-volatility, such CRP evaluation—tampering—if it is illegal relying on the fact that the resistances of memristors M_1 and M_2 are impossible to be tuned back to the original HRS value.

4.3 Proof of Electrical VP of Destruction

Set-Up phase

- The verifier prepares a memristor-PUF—the WO—and measures a number of CRPs. The verifier could secretly store either CRPs directly or a parameter model of the APUF in the DB.

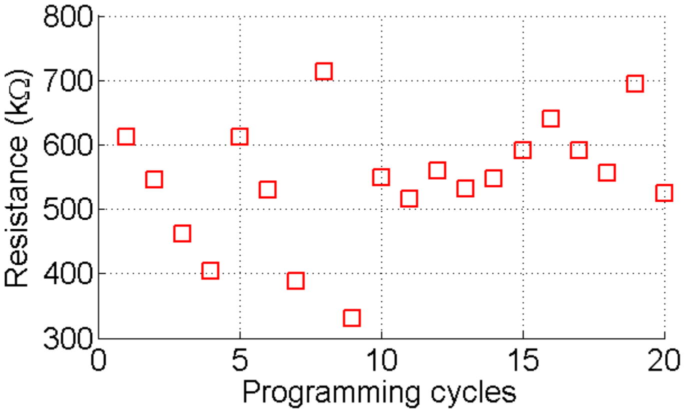

Fig. 8. Cycle to cycle (C2C) variation. The R_{OFF}/HRS variation of an individual memristor for 20 cycles, data is experimentally obtained from our fabricated memristor. A factor of two in C2C variation can be observed. Note that even larger C2C variations, eg. 50 times, have been reported [33].

- Before the memristor-PUF is sent to the prover, the verifier programs memristors into HRSs and records the resistances of M_1 and M_2 respectively.
- The memristor-PUF is transferred to the prover.

Notably, the APUF can be easily broken using machine learning techniques if the adversary collects a sufficient number of CRPs through measurements or eavesdropping.

In the memirsor-PUF—the WO, the M_1 and M_2 are treated as object OB_1, while the APUF is treated as object OB_2.

Virtual Proof The prover wants to prove that the M_1 and M_2 of OB_1 has been modified/destroyed to the verifier, and the memristor-PUF—the WO—is in his/her possession at the same time.

- The prover measures resistances of M1 and M2 by applying a small read voltage $V = V_{\mathrm{read}}$. Then the prover asks the verifier that he/she is going to prove the destruction of OB_1.
- The verifier randomly selects a set of challenges **C**s and sends them to the prover.
- The prover concatenates 1-bit responses to a multiple-bits response **R** through applying the challenges **C**s to the APUF sequentially. Then the prover sends the obtained **R** back to the server along with the measured resistances of the M1 and M2.
- The verifier firstly compares the resistances of M1 and M2 with the stored resistance value in the DB. If they are same respectively. Then the procedure continues to next step, otherwise, the VP is rejected and the VP is ended at this step.

- The verifier compares the **R** sent from the prover with stored/emulated response based on the stored CRPs or the parameter model. If they match, then the VP of the destruction is accepted, otherwise, the VP is rejected.

4.4 Discussion

The successful VP of destruction is guaranteed by the security features that the WO—memristor-PUF—has:

- The memristor-PUF cannot be cloned physically.
- The modification/destruction of original resistances of M_1 and M_2 is irreversible.
- It is impossible to obtain a number of CRPs without modifying/destroying the resistances of M_1 and M_2.

The first feature is clear, since the PUF is physically unclonable, which means even the manufacturer itself cannot forge two identical memristor-PUFs owning the same CRP behavior. The second feature is promised by the unique C2C variation inherent originated from reprogramming memristors. As for the third feature, if the prover attempts to measure a number of CRPs, the adversary has to apply $V = V_{\mathrm{pulse}}$ (positive voltage) a number of times to activate the race signal. Considering pre-setting the V_{pulse} is above the memristor threshold voltage V_{SET}, then the resistances of M1 and M2 will be inevitably changed due to decades of times of V_{pulse} applied to the APUF. Therefore, the resistances must have been changed by the prover if the **R** sent from the prover matches the response **R** in the DB of the verifier. As a fact, the modification/destruction is proved.

5 Application of VP of Destruction

In this part, we present two applications of VP of destruction. One is to secure goods supply chain, where the transfer of goods is untrusted. The other one is to secure key exchange. The key exchange builds on VP of destruction is one of approaches mitigating potential speed limitations pointed out in Section 3.3.

5.1 Secure Untrusted Supply Chain

Taking the security features of memristor-PUF in Section 4.4 into consideration, it is clear that the adversary in the supply chain can neither make a physical clone of it not build a parameter model of it without being aware by the prover and the verifier if the adversary attempts to do so. To ease the description, we treat products or goods integrated with memristor-PUFs as *physical entity.*

The hybrid memristor-PUF security primitive is able to secure supply chain, which the conventional PUF-based authentication seems unable to offer now due to the threats posed by the machine learning attacks when physical access to the physical entity integrated with a PUF to measure enough number of CRPs to build a model is possible. The physical access to the PUF in supply chain is untraceable. Our proposed memristor-PUF mitigates this threat. Furthermore, we do not have to employ a strong PUF as conventional PUF-based authentication does. Such a strong PUF requirement is to prevent fully characterization of all the CRPs within a short time once the adversary has access to it. As for the memristor-PUF, the adversary is not allowed to measure CRPs without leaving proofs that can be found by the prover and the verifier, so the memristor-PUF does not need such stringent requirement that seems hard to achieve in practice with minimized cost. We discuss the security from two parts: adversary part and prover part.

Adversary The adversary firstly cannot clone the physical entity, and secondly measure a number of CRPs to build a model as the verifier does without being noticed according to the security features listed in Section 4.4.

Notably the adversary can also measure the resistance before applying arbitrary challenges to the APUF. The adversary may try to tune M_1 and M_2 back to the original resistance value after enough number of CRPs are measured. However, this is not effective. Firstly, it is impossible to acquire the same resistance even when the adversary apples the same programming voltage as the server does due to the C2C variation. Secondly, there are two memristors. Assume the adversary attempts to tune the resistance of M_1 back to the original value by carefully adjusting. The resistance of M_2, however, will not back to the original value since the resistances of M_1 and M_2 are not allowed to be tuned separately. Carefully tuning resistance of one memristor will inevitably impact the resistance of the other memristor. Furthermore the number of memristors implemented in parallel can be increased, which will make the resistance of all of the memristors being tuned back to the original value even impossible.

Therefore, the adversary faces a major challenge to obtain the model of an APUF by collecting a number of CRPs during the shipment of the physical entity without being noticed by the prover or the server.

prover part The prover can be aware of the physical measurement of the APUF if it does happen. This physical measurement the adversary can be detected if the resistance of these memristors measured by the prover is different from the resistance recorded by the server. If no measurement occurred during shipment, the authenticity of the physical entity is promised by similarity of **R** sent by the prover to those saved in the server, where **R** is unique from one APUF from the other one given the same **C**s.

5.2 Secure Fast Key Exchange

This part we show the memristor-PUF can be utilized to secure key exchange that is fast and efficient. The key exchange protocol has following steps:

1. The verifier prepares n memristor-PUFs and builds parameter models for each of the APUFs after training a model using a number of measured CRPs. Then these n memristor-PUFs are transfered to the prover.
2. Before the key exchange operation starts, the verifier authenticates the authenticity of the n memristor-PUFs through comparing the resistances and the responses sent from the prover with the recorded values in the DB. If both of them match, the transfer of the n memristor-PUFs are secure. Up to now, it can be seen that the prover has securely got the physical memristor-PUFs, specifically APUFs, and only the verifier has the parameter model of these n APUFs.
3. The prover applies m challenges $(\mathbf{C_1}, \mathbf{C_2}, \cdots, \mathbf{C_m})$ that randomly generated by either the prover or the verifier to the O_{th}, $O_{\text{th}} \in \{1_{\text{st}}, 2_{\text{sd}}, \cdots, n_{\text{th}}\}$ memristor-PUF to obtain a concatenated response vector $(\mathbf{R_O} = r_1||r_2||\cdots||r_m)$. The verifier at the same time applies the same m challenges to all of n APUFs models and emulates $\{\mathbf{R_1}, \cdots, \mathbf{R_O}, \cdots, \mathbf{R_n}\}$.
4. The prover sends the $\mathbf{R_O}$ obtained to the verifier.
5. The verifier compares the received $\mathbf{R_O}$ with emulated $\{\mathbf{R_1}, \cdots, \mathbf{R_O}, \cdots, \mathbf{R_n}\}$ one by one. Only the emulated $\mathbf{R_O}$ will closely match to the received $\mathbf{R_O}$. This is guaranteed by the uniqueness of PUFs, which means the verifier is able to distinguish a particular PUF from a large population. Therefore, the O is the key transfered by the prover. It is clear that only the verifier is able to discover this key as only the verifier has all of these n APUF models.

The length of key can be transfered in each communication round is:

$$L_{\text{key}} = \lfloor log_2(n) \rfloor. \tag{1}$$

In other words, if $n = 2$, then $O = 1$ stands for a 1-bit digital value of '0' and $O = 2$ stands for a 1-bit digital value of '1'. As we can see, each communication round transfer a 1-bit length key.

Next we discuss the security of this key exchange protocol.

Model Building Attack The memristor-PUF prevents the adversary collecting CRPs through measurement without being ware. But the adversary is still able to eavesdrop a number of CRPs when the key exchange executes. The adversary, however, faces a major challenge to obtain a model through training it using the eavesdropped CRPs. Since the adversary has no idea which APUF generates the exposed $\mathbf{R_O}$. The fact is that the prover determines the key O, therefore, only the prover knows the key O and only the verifier is able to discover the key O as only the verifier has all models of these n APUF models. The adversary has to guess which APUF generates the $\mathbf{R_O}$ that he/she can obtain through eavesdropping. Then the adversary attempts to build a correct models based on the guessing. Considering each key exchange round exposes m CRPs of the specific APUF—eg. the third APUF—to the adversary. The number of models the adversary has to try is assumed as [34]:

$$n^{\frac{N_{\text{CRP}}}{m}} \tag{2}$$

where N_{CRP} is the number of CRPs the adversary needs to train a model that is able to predict responses for unused challenges at a specific accuracy. For example, to gain a prediction rate of 98.04% for an APUF, the adversary needs to collect 2000 CRPs—$N_{\text{CRP}} = 2000$—as shown in Table 3 as well. Therefore, the number of models the adversary has to try is $n^{\frac{1000}{m}}$. To increase the burden of computation of the adversary, XORing-APUF can be exploited. As can be seen from the Table 3, the required N_{CRP} of the XOR2-APUF is significantly increased to gain the same prediction rate of the APUF. This makes the adversary's trials become even impossible. For example, we consider $n = 64, m = 64$ and $N_{\text{CRP}} = 50000$ to gain a prediction rate of 96.09%, then the number of models becomes $64^{\frac{50000}{64}} \approx 2^{4688}$.

Table 3. Prediction Rate Comparison

	APUF		XOR2-APUF	
CRPs	P_{pred}	Time	P_{pred}	Time
500	93.71%	0:01 min	50.53%	0:01 min
1,000	96.82%	0:01 min	50.92%	0:01 min
2,000	98.04%	0:01 min	51.49%	0:01 min
5,000	99.20%	0:03 min	55.74%	0:05 min
10,000	99.60%	0:05 min	71.11%	0:34 min
20,000	—	—	84.89%	0:53 min
30,000	—	—	92.85%	1:31 min
50,000	—	—	96.09%	2:43 min
70,000	—	—	97.06%	3:33 min
100,000	—	—	97.38%	6:10 min
200,000	—	—	97.93%	12:59 min

6 Conclusion

In this paper, we extend the new security concept of VP proposed by Ruhrmair *et al.* to secure key exchange and secure goods supply chain. In addition, we propose a highly secure hybrid hardware security primitive—memristor-PUF that exploits the static variations of PUF and dynamic variations of memristor. The memristor-PUF is demonstrated to satisfy the requirement of VP of destruction, which is the first construction of VP of destruction based on electrical circuit.

References

1. U. Rührmair, J. Martinez-Hurtado, X. Xu, C. Kraeh, C. Hilgers, D. Kononchuk, J. J. Finley, and W. P. Burleson, "Virtual proofs of reality and their physical implementation," in *36th IEEE Symposium on Security and Privacy*, 2015, DOI: 10.1109/SP.2015.12.

2. K. Rosenfeld, E. Gavas, and R. Karri, "Sensor physical unclonable functions," in *Proc. IEEE. Int. Symp. Hardware Oriented Hardware-Oriented Security and Trust (HOST)*, 2010, pp. 112–117.
3. B. Fisch, D. Freund, and M. Naor, "Physical zero-knowledge proofs of physical properties," in *Advances in Cryptology–CRYPTO*. Springer, 2014, pp. 313–336.
4. K. Eshraghian, O. Kavehei, K.-R. Cho, J. M. Chappell, A. Iqbal, S. F. Al-Sarawi, and D. Abbott, "Memristive device fundamentals and modeling: applications to circuits and systems simulation," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1991–2007, 2012.
5. D. Lim, "Extracting secret keys from integrated circuits," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.
6. G. E. Suh and S. Devadas, "Physical nclonable functions for device authentication and secret key generation," in *Proceedings of the 44th Annual Design Automation Conference*. ACM, 2007, pp. 9–14.
7. M. Potkonjak and V. Goudar, "Public physical unclonable functions," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1142–1156, 2014.
8. U. Rührmair, "Simpl systems: On a public key variant of physical unclonable functions." *IACR Cryptology ePrint Archive*, vol. 2009, p. 255, 2009.
9. B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, 2002, pp. 148–160.
10. B. Gassend, D. Lim, D. Clarke, M. Van Dijk, and S. Devadas, "Identification and authentication of integrated circuits," *Concurrency and Computation: Practice and Experience*, vol. 16, no. 11, pp. 1077–1098, 2004.
11. J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *Proc. IEEE Symp. VLSI Circuits, Digest of Technical Papers*, 2004, pp. 176–179.
12. D. Suzuki and K. Shimizu, "The glitch PUF: A new delay-PUF architecture exploiting glitch shapes," in *Cryptographic Hardware and Embedded Systems–*CHES. Springer, 2010, pp. 366–382.
13. D. E. Holcomb, W. P. Burleson, and K. Fu, "Initial SRAM state as a fingerprint and source of true random numbers for RFID tags," in *Proceedings of the Conference on RFID Security*, vol. 7, 2007.
14. Holcomb, Daniel E, W. P. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1198–1210, 2009.
15. Y. Su, J. Holleman, and B. P. 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, 2008.
16. R. Maes, P. Tuyls, and I. Verbauwhede, "Intrinsic PUFs from flip-flops on reconfigurable devices," in *3rd Benelux Workshop on Information and System Security (*WISSec), vol. 17, 2008.
17. V. van der Leest, G.-J. Schrijen, H. Handschuh, and P. Tuyls, "Hardware intrinsic security from D flip-flops," in Proceedings of the fifth ACM Workshop on Scalable trusted computing, 2010, pp. 53–62.
18. S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "The butterfly PUF protecting ip on every FPGA," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 67–70.
19. M. Roel, "Physically unclonable functions: Constructions, properties and applications," Ph.D. dissertation, University of KU Leuven, 2012.

20. C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of IEEE*, vol. 102, pp. 1126–1141, 2014.
21. L. Zhang, Z. H. Kong, C.-H. Chang, A. Cabrini, and G. Torelli, "Exploiting process variations and programming sensitivity of phase change memory for reconfigurable physical unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, pp. 921–932, 2014.
22. Y. Gao, D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott, "mrPUF: A novel memristive device based physical unclonable function," in *13th International Conference on Applied Cryptography and Network Security*, 2015.
23. Y. Gao, D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott, "Memristive crypto primitive for building highly secure physical unclonable functions," *Scientific Reports*, vol. 5, art. no. 12785, 2015.
24. L. Zhang, X. Fong, C.-H. Chang, Z. H. Kong, and K. Roy, "Highly reliable spin-transfer torque magnetic RAM based physical unclonable function with multi-response-bits per cell," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1630–1642, 2015.
25. Y. Gao, D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott, "Emerging physical unclonable functions with nanotechnology," IEEE Access, 2015, In press.
26. U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, 2010, pp. 237–249.
27. Y. Gao, D. C. Ranasinghe, G. Li, S. F. Al-Sarawi, O. Kavehei, and D. Abbott, "A challenge obfuscation method for thwarting model building attacks on PUFs," Cryptology ePrint Archive, 2015, https://eprint.iacr.org/2015/471.pdf.
28. Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, and U. Ruhrmair, "The bistable ring PUF: A new architecture for strong physical unclonable functions," in *Proc. 2011 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2011, pp. 134–141.
29. R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.
30. A. Maiti, J. Casarona, L. McHale, and P. Schaumont, "A large scale characterization of RO-PUF," in *IEEE International Symposium on Hardware-Oriented Security and Trust (*HOST),, 2010, pp. 94–99.
31. D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 10, pp. 1200–1205, 2005.
32. A. Chen, "Utilizing the variability of resistive random access memory to implement reconfigurable physical unclonable functions," *IEEE lectron Device Letters*, vol. 36, no. 2, pp. 138–140, 2015.
33. A. Chen and M.-R. Lin, "Variability of resistive switching memories and its impact on crossbar array performance," in *IEEE International Reliability Physics Symposium (IRPS)*, 2011, DOI: 10.1109/IRPS.2011.5784590.
34. M. Rostami, M. Majzoobi, F. Koushanfar, D. S. Wallach, and 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, 2014.