# Scalable Password-Changing Protocol for Smart Grid Device Authentication

Rehana Tabassum, Klara Nahrstedt, Edmond Rogers, Michael Rawson
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
Email: {tabassu2, klara, ejrogers, mr4}@illinois.edu

***Abstract*- In smart grid, the scale of pole devices that monitor the health of power lines is already large, and with the upgrade of the smart grid, the number of these resource-constrained devices is further increasing. These devices are easy targets to security attacks due to wireless access communication and due to weak passwords used to access and read telemetric data by the pole maintenance personnel.**

**In this paper, we present a scalable and automated password-changing protocol framework for unique authentication of human personnel (driver) with large scale of pole devices, and for secure collection of telemetric data from the pole devices. Our protocol framework employs physical per-driver, per-pole-device information as well as fractal functions to generate new unique passwords and secrete keys for different drivers over a large scale of pole devices and over long periods of time. Our experiments confirm that the password-changing protocol authenticates and transmits pole device data securely and in real-time under varying maintenance scenarios.**

## I. INTRODUCTION

The current power grid system and its power lines in the field are monitored by telemetric devices, which are sensors with capacitor banks and are placed on top of electric poles. They measure frequency, voltage and current readings from power lines, which need to be maintained in the field. The maintenance personnel (drivers) from utility companies collect data readings from these telemetric devices to their handheld devices on a regular basis to ensure that the health of power line is sound and stable as shown in Figure 1. Also, when damage occurs due to any kind of natural disasters (e.g., storm), the utility companies find out the faulty location by analyzing the unusual data readings taken from these devices.

In the current power grid system, **security of data** inside of the telemetric and handheld devices is one important concern. The telemetric and handheld devices and their data are easy target to security attacks due to the **wireless communication** over which data are read from/to the telemetric device to/from the handheld device of the driver (maintenance personnel), and due to the **weak passwords and vulnerable authentication protocol** that utilities use to access the handheld and telemetric devices. Especially with the increased scale of these small resource-constrained devices, due to on-going upgrade of smart grid, the security threats are further increasing.

Therefore, the goal in this paper is to investigate a robust, scalable password-changing protocol framework to ensure device authentication and secure access to data inside the handheld and telemetric resource-constrained devices along with the secure delivery of data over the wireless network in the field as shown in Figure 1.

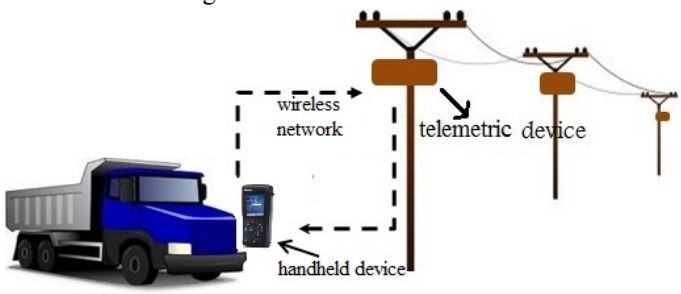


Fig. 1. In- field Scenario

Password verification problem over an insecure network has been researched for very long time. There are many existing solutions for solving password based authentication problem over an insecure network. Public-key based cryptography is a fundamental and widely used solution. Gong et al. introduced an asymmetric PKI model in [8], where, in addition to a password, the user uses the public key of the server. Later Halevi and Krawczyk [9] have given formal definitions and proofs of security in this setting. Another setting for this problem is one where two parties share only one password and neither party knows each other's public-key. This setting was first considered by Bellovin and Merritt in [2] which is known as Encrypted Key Exchange (EKE) protocol. They implemented EKE protocol using RSA [3] and Diffie–Hellman [4] asymmetric cryptosystems. Complexity of these RSA and Diffie-Hellman cryptosystem relies on mathematical relationships, i.e., integer factorization and discrete logarithmic problems, respectively. Protocols such as SPEKE [11], DHEKE [5], A-EKE [7], SRP [6] have been proposed in later time which are stronger protocols of the EKE family. However, these public-key based authentication mechanisms

incur heavy computation and memory overhead during the processes of signature generation and validation [13]. On the contrary, in symmetric-key cryptography, identical keys are used for both encryption and decryption and each trading partner can use the same publicly known encryption algorithm; no need for developing and exchanging secret algorithms here. Although they are much simpler and faster than these public-key based authentication mechanisms, they are not as strong as public-key cryptography. The main drawback of symmetric encryption is that a shared secret key must be agreed upon by both parties prior to any communication in a very secure manner. In addition, authenticity of the origin cannot be proved when the secret key is shared among multiple devices. However, in our approach we use both public-key cryptography and symmetric-key cryptography to achieve better performance.

In this paper, we propose a fast, cost-effective, scalable, and robust password-changing protocol framework, which ensures generation of strong passwords to access different pole telemetric devices in different maintenance situations. We generate device passwords to be used for authentication between handheld and telemetric devices and shared keys to be used for secure data communication, using **physical information** such as local time, pole location, device ID, and user ID, and **fractal functions**. Our protocol framework ensures that device passwords and shared exchange keys are changing each time a operator (maintenance personnel) accesses a telemetric device and its data using his/her handheld device along his/her route, and data are transmitted in secure and real-time manner. Our analysis and results confirm the claims.

The rest of this paper is organized as follows. In section II the model and set of assumptions considered are defined. Section III describes the details of the proposed scheme. We carry out the security analysis of the proposed scheme in section IV. Section V addresses performance and implementation issues. We conclude in section VI.

## II. SYSTEM MODELS AND ASSUMPTIONS

### A. Network Model

In the system, sensor and capacitor banks placed on electric poles measure telemetric measurements from power line and store it as shown in Figure 1. A radio is attached underneath the capacitor banks; this radio is used to get the stored data readings. For ease of the reading, we will consider two devices, a telemetric device that produces data measurements from the capacitor banks and a telemetric device-reader as a handheld device throughout this paper. A point-to-point radio (wireless) network is established between the handheld device and the telemetric device during communication of these two parties. The standard we use in our validation is *IEEE 802.11,* however other wireless standards such as IEEE 802.15.4 (Zigbee) can be used.

### B. Data Model

Usually, telemetric devices collect telemetric measurements of frequency, voltage, current readings of power lines. These telemetric measurements are sent from the telemetric device to the driver's handheld device over the wireless network in small packets.

### C. User Security Model

We assume that operator can memorize a **password**, serving as a user password, and a **PIN** (Personal Identification Number), serving as a user name, for authentication. The password and PIN are given to the operator by the utility. With this information, he/she logs into the handheld device (Note: The functionalities of handheld device are built into the car, hence the driver (maintenance personnel) may login with password and PIN once, when he/she starts driving, and not at each pole.) The list of PINs and corresponding passwords (in hashed form) is saved in the handheld device memory. Our protocol assumes that there is a trusted setup phase at the utility site, prior to any communication, in which the employee's password and PIN information is stored (in hashed format) in a database of the handheld device memory.

In addition, necessary crypto functions such as mixing function, key derivation fractal function, AES symmetric key function; crypto algorithms such as iterated block cipher, are agreed upon and installed on handheld and telemetric devices (Note: Installation or refresh configuration of functions/keys on telemetric devices is critical, but out of scope in this paper.) Symmetric key encryption is used, i.e., a shared secret key is used for both encryption and decryption of the packets to send them over the wireless network. For current scenario, Advanced Encryption Standard (AES) is used since it is fast in both software and hardware and performs well on a wide variety of hardware.

### D. Attack Model

Since the whole system exists in an open environment, security barrier to prevent unauthorized access from outside of the domain is very necessary. An attacker may try to get access of the devices by faking identities if the attacker gets the common password. Since the network is wireless, attacker may get detailed information easily by eavesdropping on the communications. Even worse, attackers may physically capture telemetric devices, read cryptographic information out of memory and exaggerate their infections. Under cover of faked

identities, they can perform various types of attacks, like denial of service attack, bad data injection and others. A detailed security analysis is provided in section IV.

## III. Approach

This section presents our password-changing protocol that provides robust authentication and secure communication. We divide our approach in three phases:

- Phase 1: Authentication protocol of a maintenance personnel to the handheld device,
- Phase 2: Authentication protocol between the handheld device and the telemetric device and
- Phase 3: Secure communication protocol between the telemetric device and the handheld device

As mentioned earlier, the set of activities on phase 1, i.e., authentication of the maintenance personnel to the handheld device, needs to be performed once, when s/he starts driving for collecting data. Whereas, the handheld device needs to authenticate itself to each telemetric device with a unique password at each pole location and collect data readings afterwards. Detailed steps of these phases are described as follows. Necessary mathematical notations of the symbols used in this section are given in Table 1.

### *A. Phase 1*

All methods of human authentication fall into three broad categories [6]:

- the knowledge factors: Something the user knows (e.g., password, pass phrase, PIN, response to a challenge)
- the ownership factors: Something the user has (e.g., ID card, security token, software token, cell phone)
- the inherence factors: Something the user is or does (e.g., fingerprint, retinal pattern, DNA sequence, signature, face, voice)

In our approach, we consider mostly the first category since it is easier to use, convenient and less expensive to deploy, than token-based or biometric methods. This phase deals with the authentication of OP to HH, yet maintaining the integrity of OP. The authentication process consists of three steps:

1) Entering user name, password
2) Solving challenge-response query or
3) Responding to CAPTCHA query
    i. Text-based CAPTCHA query or
    ii. Image based CAPTCHA query or
    iii. Utility code verification

The first step ensures integrity of an operator while collecting data at poles along his/her route. The second step ensures the uniqueness of the OP by challenge-response. Finally the third step ensures that actually a human enters the above two steps of information and not a robot.

***Step 1:*** The operator OP enters his/her PIN and a common password into HH. Upon receiving PIN and OP-password, HH verifies the PIN, OP-password combination. PIN, which is a low-entropy number, enables the system to verify the integrity of OP, i.e., it keeps track, which OP is logging in. The reason is that it is always easier for an insider to break the system than an outsider, and hence, we used a way to identify the responsible person.

***Step 2:*** In the next step, OP performs a challenge-response query by entering operator's secret question answer $SQ_A$. Since many people put a lot of their private information on social networking sites, it is easy for an attacker to find the answer to any private question such as maiden name, date of birth and others. Therefore, instead of using the traditional challenge-response approach, the query should be rather employment or company specific i.e., something that only employee knows the answer. These secret question answers are different for each operator, are chosen prior to the data collection and stored in the handheld device in hash forms in the trusted setup phase. This step is included to ensure that even if HH is stolen, the intruder cannot have access to HH to get access to the TD.

TABLE I
Mathematical Notations

| Symbol | Definition |
|---|---|
| *OP, HH, TD* | System Principals (Operator, Handheld Device, Telemetric Device) |
| *PIN* | Personal Identification Number |
| $SQ_A$ | Secret Question/Answer |
| $challenge_H$ | Challenge created by HH |
| $H()$ | One-way hash function |
| $E_{PUi}()$ | Encrypt operation with Public Key of *i*th telemetric device |
| $D_{PRi}()$ | Decrypt operation with Private Key of *i*th telemetric device |
| $E_{P''}()$ | Encrypt operation with secret key, *p''* |
| $P$ | Session Shared Key of 128 bit |
| $p'$ | *k* Most Significant Bits(MSB) of *P* |
| $p''$ | (128-*k*) Least Significant Bits of *P* |
| $k$ | Number of MSB of *P* to verify |
| $S_{cur,L}$, $S_{prev,L}$ | Salt (current and previous) at Location L |
| $r$ | Random number in range [3,4) |
| $t$ | Time (time stamp) |
| $HD_{id}$ | Handheld Device id |
| *ACK* | Acknowledgement |
| *ERR* | Error message |
| *TER* | Terminate message |
| $Q()$ | Iterated Block Cipher Algorithm |
| $f()$ | Mixing function |
| $i$ | Integer |
| *RAND* | 128-bit random challenge |
| $M_{sign}$ | Message *M* is signed by the private key of the telemetric device |

One way hash function, *H()* is used to make sure that even if a HH is stolen, i.e., the database of HH is compromised, an intruder cannot get OP's password file or secret question answers, impersonate as a legitimate operator and use the device. HH looks up the *PIN* entry and fetches the stored hash value of secret question answer from memory. It also computes the hash of received value $H(SQ_A)$ and verifies if this calculated value matches with the stored one.

***Step 3:*** When the handheld device finds a valid *PIN*, it starts collecting physical information, i.e., GPS location, temperature, humidity, time *t* and handheld device ID ($HD_{id}$). Here *device ID,* $HD_{id}$ is some unique physical information (e.g., 48 bit MAC address) of the HH. Timestamp *t, PIN* and handheld device ID are used to calculate *P* (session shared key) later in phase 2. In this phase, HH combines location, temperature, humidity and time to create a CAPTCHA automatically. CAPTCHA is a program that generates and grades tests that are human solvable, but beyond the capabilities of current computer programs [10]. Thereby, in our approach, CAPTCHA is used to protect handheld devices against remote software robots to get access to the telemetric device. In [12] different types of CAPTCHAs are presented such as text-based, audio, image-based and others. In our protocol, an OP can choose to respond to either text-based or image-based CAPTCHA; or he can verify himself entering the code given by utility company. The use of physical information to generate random CAPTCHAs results in different CAPTCHA each time and thus increases the strength of this phase. Fig. 2 formalizes the above procedure. The GUI of this section is shown in Fig. 2.

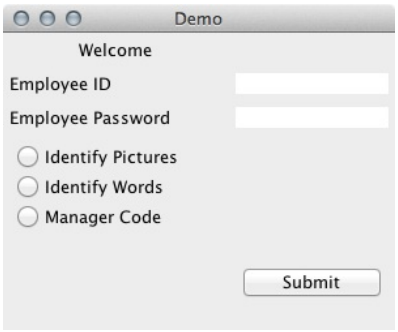


Fig . 2: GUI for the maintenance personnel with options

Fig. 3 formalizes the above procedure of a robust authentication of phase 1. Now if the PIN and OP-password, inserted by operator, are valid and he can answer CAPTCHA correctly, (which indicates a person with a valid PIN is present at the location) the integrity of the OP is assured. In the next phase, it sends the **login request message** to the TD.

| OP | HH |
|---|---|
| 1. Enter PIN and Password | |
| | 1. Verify PIN, Password |
| | 2. Show Question $SQ_A$ |
| 2. Enter Answer $SQ_A$ | |
| | 2. Check added $SQ_A$ |
| | 3. Collect Physical Information (e.g., location, temperature, Time t, $HD_{id}$) |
| | 3. Generate CAPTCHA |
| 3. Enter Answer of CAPTCHA | |
| | 3. Verify CAPTCHA Answer |

Fig. 3: Phase 1 - Authentication of operator

*B. Phase 2*

In this phase, when the OP comes within the wireless network, the HH sends *PIN*, *t* and $HD_{id}$ (collected in phase 1) in the form of **login request message**, $m_1$. The handheld device encrypts the message $m_1$ with the public key $PU_i$ of the TD *'i'*. The HH transmits the encrypted message to the TD over the wireless network to initiate a conversation with the TD. The procedure for this phase is shown in Fig. 4.

At this point, the telemetric device gets driver's (maintenance personnel) PIN, timestamp (at which the driver logged in to the handheld device) and handheld device ID by decrypting the received message using its own private key. TD then generates a random number and transmits it to handheld device after signing it with its private key, to ensure the authenticity of the message. This protects against man-in-the-middle attack. Both devices start calculating password, *P* using following equation: $P = Q\ (\ PIN,\ S_{cur,\ L},\ RAND\ )$. Here, *Q()* is an iterated Block Cipher Algorithm and $S_{cur,\ L}$ is a salt combined with the *PIN* and *RAND* to make a dictionary attack expensive.

In our approach, a fractal function *S* is used to calculate salt value, $S_{cur,\ L}$. Current salt value, $S_{cur,\ L}$ of location *L* depends on the previous salt value, $S_{prev,\ L}$ that is stored at TD of location *L*. Both devices calculate $S_{cur,\ L}$ by the following equation: $S_{cur,\ L} = r * S_{prev,\ L} * (\ 1\ -\ S_{prev,\ L})$.

This fractal function has the property that for a particular value of *r* in the range [3,4), the salt values repeat over an interval, e.g., for *r* = 3.2, four values of $S_{cur,L}$ are different and appear random to an attacker, but after the forth value, repetition of the four values occurs. For other values of *r*, the repetition interval is different. In our approach, both devices generate *r* using a mixing function *f()* taking the inputs *device ID*, and time *t* using the equation: $r = f\ (\ HD_{id},\ t\ )$.

Here *device ID,* $HD_{id}$ is some unique physical information (e.g., 48 bit MAC address) of the handheld device, HD. The reason for using physical information to calculate *r* is to giving insufficient information (the attacker does not have device ID

or timestamp, *t*) to an outside intruder to mount dictionary attack against the password, *P*.

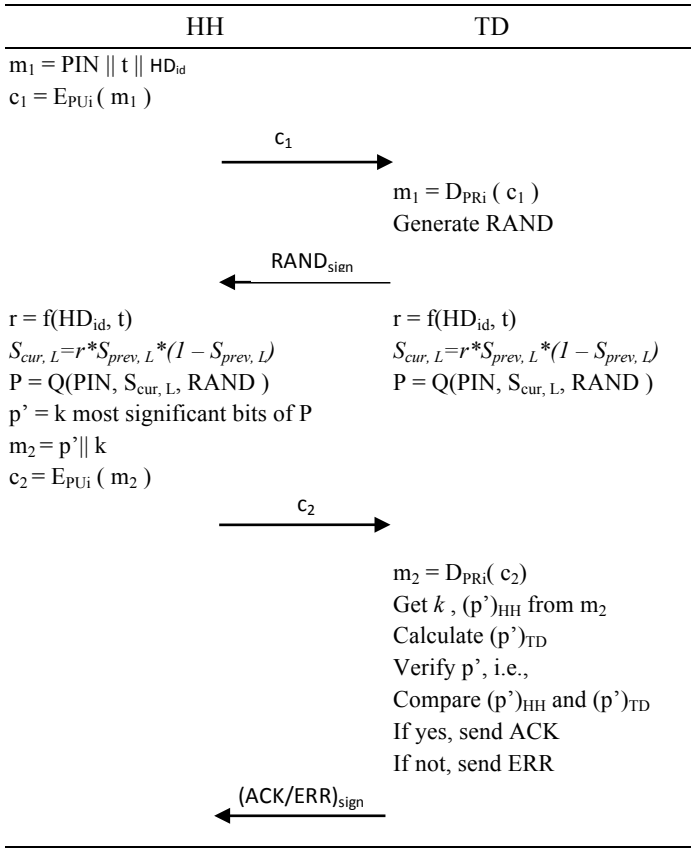


Fig. 4. Phase 2 - Authentication of handheld device

In our password changing protocol, only *k* most significant bits (MSB) of *P*are used as device password (*p'*) for the authentication of HH to TD (different from the OP's common password entered in phase 1). Rest of the bits of *P* is used as symmetric key (*p''*) for secure communication of data between HH and TD in phase 3. HH chooses *k* randomly, appends the value of *k* with *p'*, encrypts it with the public key of TD and transmits the encrypted message to the TD as the computed response. Since only that particular TD has the private key, no other device can decrypt the message. Upon receiving the encrypted message, $c_2$, the telemetric device decrypts it by: $m_2 = D_{PRi}( c_2)$. The telemetric device checks for the received and calculated *k*-bit values. If the received and calculated *k*-bit values do not match with each other, the authentication is failed and an error message, ERR is sent. Otherwise, an acknowledgement, ACK is sent to the HD. This ERR or ACK message is sent after signing it with the private key, so that the handheld device knows that this message comes from the telemetric device but from any other devices. In this way, authenticity of the message is maintained by signing every message from TD with it's own private key ($PR_i$) and confidentiality of the message is maintained by encrypting every message from HH with the public key ($PU_i$) of TD. Thus in this phase, the authentication of the handheld device to the telemetric device is accomplished.

### *C. Phase 3*

In this phase, secure delivery of the telemetric data is ensured. Both devices use a symmetric key encryption and the symmetric key *p''* is the *(128 – k)* bits of *P* derived in phase 2. The telemetric device reads the telemetric measurements from memory and encrypts that data with the calculated key, *p''* by: $d = E_{P''}( data )$. The telemetric device sends *d* to handheld device over the wireless network.

Upon receiving the encrypted data, handheld device decrypts it with *p''* by: $data = D_{P''}( d )$ and stores the data on handheld device in secure database. Finally, they conclude when the telemetric device sends a TD-signed termination message, TER, to ensure that the session is terminated. Fig. 5 shows the details.

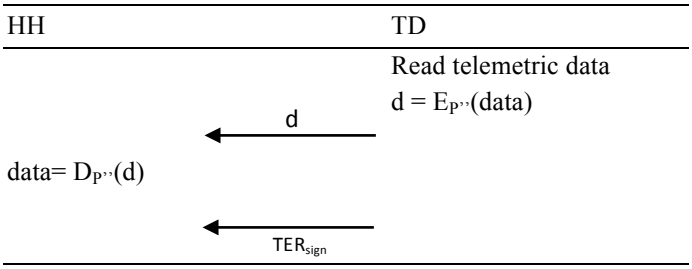


Fig. 5: Phase 3 - Communication between two devices

## IV. SECURITY ANALYSIS

Our approach correctly generates a valid password at the handheld device if OP supplies a correct credential (i.e. PIN, password) and the software on both sides performs correctly. However, it is more difficult to show that our protocol is secure. In this section we analyze the security of our password changing protocol against different types of attacks.

***Replay Attack:*** Our protocol does not use a stored password for authentication of HH; rather HH generates unique password, *p'* instantly. So, an attacker cannot use password of current session in any future sessions to authenticate a device to TD. Thus our protocol thwarts *replay attack*.

***Forward secrecy:*** If the intruder guesses a password during one successful run, it should not allow him to determine the password of past sessions–known as *forward secrecy*. If an intruder guesses *p'* from overhearing $c_2$, s/he cannot reproduce *P* for future sessions since salt is not exchanged but stored in TD and HH. Thus Our protocol maintains *forward secrecy*.

***Dictionary attack:*** In our protocol, important messages ($m_1$, $m_2$) are encrypted before transmitting. This prevents the attacker from being able to guess necessary parameters based

on exchanged messages to calculate and verify $P$. We use physical information ($HD_{id}$, $t$) to calculate $r$ to give insufficient information to an intruder to mount dictionary attack against $P$. First of all, the attacker does not have device ID or timestamp, $t$. In addition, the value of $r$ is generated each time immediately prior communication. Also, by using salt our protocol makes it slower for an intruder to mount dictionary attacks.

***MITM Attack:*** A man-in-the-middle (MITM) attack, which requires an attacker to fool both sides of a legitimate conversation [6], is not possible in our protocol. An intruder can sit in the middle of handheld device HH and telemetric device TD, block the messages ($c_1$, $c_2$, $RAND_{sign}$, $ACK/ERR/TER_{sign}$) sent by HH or TD and send garbled messages acting like either HH or TD. Since, every message from TD is signed by the private key ($PR_i$) and all messages from HH are encrypted by the public key ($PU_i$) of TD, assuming that the private key is not compromised, the intruder cannot sign or decrypt the messages. Thus, our protocol protects against MITM attack.

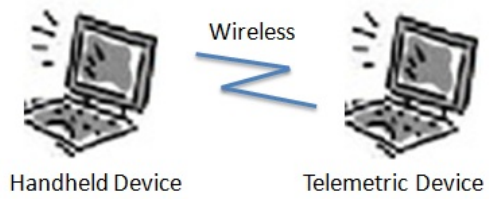


Fig. 6. Experimental Setup

## V. Implementation and Evaluation

The implementation setup is shown in Fig. 6. One laptop is used as the handheld device of the operator and another laptop is used as the telemetric device having telemetric readings. For both laptops, we used Intel Core 2 Duo 2.26GHz processors with 2GB read-only memory. The prototype is implemented in java so that it can be easily ported into the mobile devices. The communications between the laptops are performed using wifi. AES symmetric key cryptography is used for encryption and decryption of the shared key (in phase 2) and the telemetric readings (in phase 3).

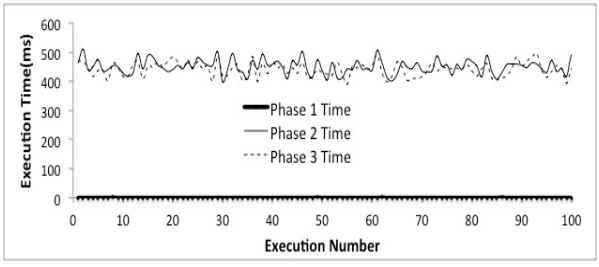


Fig. 7. Execution time of password changing algorithm in three phases for different executions

Fig. 7 shows the execution time of password changing algorithm in different phases for different executions. Since, the authentication process of operator in Phase 1 is only responsible for challenge-response, generating and verifying CAPTCHAs locally at the handheld device; the process requires very low time and space complexity. Also, the operator side delay is considered negligible here. Therefore, the execution time in Phase 1 is very small. However, the execution times in Phase 2 and Phase 3 are large and lie in the range of 400 - 500 ms each. The round trip time from and to handheld device and telemetric device and encryption-decryption time is the main reason for the large value of execution time in phase 2 and phase 3.

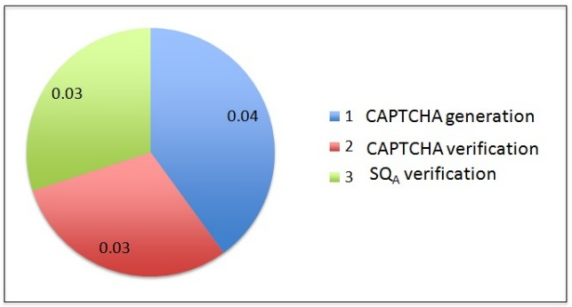


Fig. 8(a). Execution time (ms) in phase 1

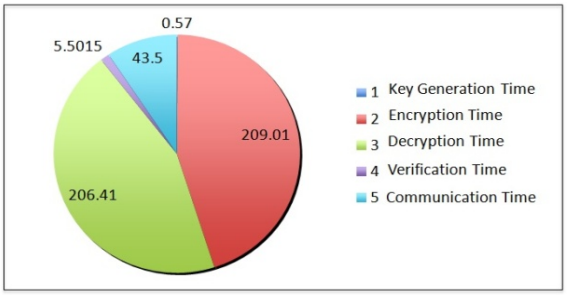


Fig. 8(b). Execution time (ms) in phase 2

To understand the execution time of different steps in each phase, we represent pie charts in Fig. 8 dissecting the average execution time of each phase. Fig. 8(a), 8(b) and 8(c) shows

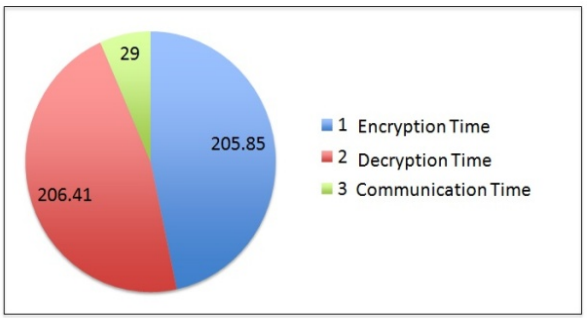


Fig. 8(c). Execution time (ms) in phase 3

three different pie charts for phase 1, phase 2 and phase 3 respectively. In phase 1, solving challenge-response query, generation of CAPTCHA and verification of CAPTCHA take almost equal time (Fig. 8(a)). In phase 2, the average execution time is 450.4915ms. The protocol takes very small amount of time (0.57ms) for key generation. The rest of the time is taken mainly on encryption-decryption along with wireless communication between two laptops. In Fig. 8(c) the execution time for each step for phase 3 is shown. The AES encryption and decryption of the shared key (in phase 2) and the telemetric readings (in phase 3) take most of the time. On the other hand the key generation and exchange of key takes very small amount of time.

## VI. CONCLUSIONS

In this paper, we highlight one of the authentication problems in the smart grid critical infrastructure; the weak access control and vulnerable authentication/secure transmission protocol to telemetric pole devices in the field. Our password-changing protocol framework achieves creating passwords and shared keys based on physical characteristics such as per-pole-device locality, per-pole temporal and per-driver identifications (e.g., *PIN, $HD_{id}$*). The experimental results show that our scheme is computationally efficient, and yields strong security for large number of TD devices in varying maintenance scenarios.


## ACKNOWLEDGMENT

This research is supported by the Department of Energy under Award Number DEOE0000097.


## REFERENCES


[1] W. Diffe and M.E. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory, IT-22(6): 644 654, November, 1976.

[2] S. M. Bellovin, and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks," *Proceedings of the I.E.E.E. Symposium on Research in Security and Privacy,* Oakland, May 1992.

[3] R.L. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Communications of the ACM, vol 21, pp. 120-126,* Feb 1978.

[4] W. Diffie and M. E. Hellman, Privacy and Authentication: An Introduction to cryptography, Proceedings of the I.E.E.E., vol. 67, No. 3, pp. 397-427, March 1979.

[5] M. Steiner, G. Tsudik, and M. Waidner, Refinement and extension of encrypted key exchange, ACM SIGOPS, vol. 29, no. 3, July 1995.

[6] T. Wu**,** The secure remote password protocol, *Internet Society symposium on Network and Distributed System Security*, 1998.

[7] S. M. Bellovin and M. Merritt, Augmented Encrypted Key Exchange: a Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise, *AT&T Bell Laboratories,* 1994.

[8] L. Gong, M. Lomas, R. Needham, and J. Saltzer, Protecting Poorly Chosen Secrets from Guessing Attacks, *IEEE Journal on Selected Areas in Communications*, vol. 11, No. 5, pp. 648–656, 1993.

[9] Shai Halevi, and Hugo Krawczyk, Public-Key Cryptography and Password Protocols, *ACM Trans. on Information and Systems Security,* vol. 2, No. 3, pp. 230–268, 1999.

[10] L. von Ahn, M. Blum, and J. Langford, "Telling Humans and Computers Apart (Automatically) or How Lazy Cryptographers do AI," *Communications of the ACM, vol. 47, no. 2, pp. 57-60,* 2004.

[11] D. P. Jablon, Strong Password-Only Authenticated Key Exchange, *ACM Computer Communications Review*, October 1996.

[12] E. Bursztein, S. Bethard, C. Fabry, J. Mitchell, and D. Jurafsky, How Good are Humans at Solving CAPTCHAs? A Large Scale Evaluation, *IEEE Symposium on Security and Privacy,* 2010.

[13] Y. Huang, W. He, and K. Nahrstedt, ChainFarm: A Novel Authentication Protocol for High-rate Any Source Probabilistic Broadcast, *IEEE*, 2009.