SCAPACH: Scalable Password-Changing Protocol for Smart Grid Device Authentication

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

King-Shan Lui
Department of Electrical and Electronic Engineering
The University of Hong Kong
kslui@eee.hku.hk

Abstract— **In smart grid, the scale of pole devices that monitor the health of power lines is already large, and with the upgrade of smart grid, the number of these resource-constrained (in terms of memory and computation) devices is further increasing. These devices are easy targets to security attacks as they are accessible via wireless network, and they use weak passwords for authentication and reading telemetric data by the pole maintenance personnel.**

In this paper, we present a SCalable and Automated PAssword-CHanging protocol, SCAPACH, for unique authentication of human personnel (operator) with large scale of pole devices, and for secure collection of telemetric data from the pole devices. SCAPACH employs physical per-operator, per-pole-device information as well as changeable secret salts to generate new unique passwords and secret keys every time a pole device is accessed. 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

Current power grid systems and their power lines in the field are monitored by telemetric devices (TD), 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 (operators) from utility companies collect data readings from these TDs to their handheld devices (HD) on a regular basis to ensure that the health of power line is sound and stable (Figure 1). These data are critical when damages occur due to any kind of disasters and the utility company needs to identify the faulty location by frequently analyzing the unusual data readings taken from these TDs.

In current power grid systems, **security of data** inside TD and HD is an important concern. TDs and HDs, and their data are easy target to security attacks due to the **wireless channel** over which data are transmitted, and also due to the **weak passwords and vulnerable authentication protocol** that utilities use to access theses devices. TDs are typically secured by simple passwords, known to many users (operators), with the same password often used for a large number of devices. Besides, telemetric measurements are transmitted over wireless channel encrypted by the same symmetric key stored in both devices every time. The security threats are further increasing with the increased scale of these small resource-constrained devices due to continual security reviews and cryptanalysis advancements [2]. Therefore, the development of a robust, scalable password-changing protocol framework is imperative to ensure secure device authentication and secure delivery of data within real-world constraints.

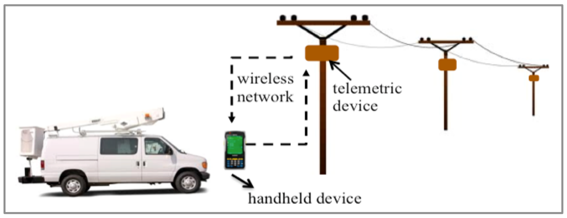


Fig. 1. In-field Scenario

Password verification problem over an insecure network has been investigated for a long time. Many existing security solutions have been built based on Diffie-Hellman (DH) key exchange protocol. In 1992, Bellovin and Merritt [1] proposed Encrypted Key Exchange (EKE) protocol, which is a password-authenticated key agreement method, based on RSA and DH. Protocols such as SPEKE [8], DHEKE [4], A-EKE [6], and SRP [5] have been proposed in later time, which are strongly secured protocols of the EKE family. These approaches are computationally expensive; and the number of messages exchanged between the two parties is also not trivial. However, since the TDs have limited storage and computational capacity, we need lightweight (in terms of memory and computation overhead) protocols for TD to perform associated cryptography [13].

Key management is another important issue in security. Traditional PKI systems (e.g., X.509) are not used in smart grid due to their structural complexity and cost for establishing and managing the framework. As compared to the PKI systems, Identity-Based Cryptography (IBC) is much simpler [11]. Shamir [11] introduced the concept of IBC and since then many ID-based key agreement protocols have been proposed. In [12], IBC-based cryptography system is used for communications in smart grid networks, where machine identification number of a device is used to generate unique keys. Not only this scheme is computationally expensive but also it requires the modification inside each TD (i.e., the memory of pole-top TDs needs to be reconfigured) when a new HD is added. This approach is not feasible in our scenario because of the limited change management capabilities [2] of the TDs. Therefore, we address the limited change management capability problem as well as the memory and computational constraint problem of TDs in our solution.

In this paper, we propose a fast, cost-effective, scalable, and robust password-changing protocol framework, SCAPACH, which generates new device passwords to be used for **authentication between handheld and telemetric devices**, and symmetric keys to be used for **secure data communication**. To the best of our knowledge, this is the very first attempt to address our goals. We introduce *Physical Unclonable Functions (PUFs)* to alleviate the load of TDs in generating and keeping keys without revealing them. Thus we lessen the memory and computational burden from TDs. SCAPACH generates device passwords and symmetric keys based on physical information (such as local time, pole geographical location, handheld device id etc.) and changeable stored secret; hence, they are short-lived. We ensure that 1) different device passwords and symmetric keys are generated inexpensively and used every time an operator accesses a TD using his/her HD, and 2) data are transmitted in a secure and real-time manner. Our analyses and implementation results confirm the claims.

II. SYSTEM MODELS AND ASSUMPTIONS

A. Network Model

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

B. Data Model

Usually, TD collects telemetric measurements of frequency, current, voltage readings of power lines and stores these measurements. According to the utility companies [10], these telemetric measurements are not sent over PLT (Power Line Telecommunication) due to the large amount of data. They are sent from the TD to the operators' HD over the wireless network in small packets. Intruders may get unauthorized access and change the telemetric measurements maliciously at the TD, which may lead to wrong decision-making and false situational awareness for utilities. Therefore, securing the access of devices (both TD and HD) and their communication channel is important for the utility companies.

C. User Security Model

We assume that the operator (OP) can only access a HD if s/he has a unique identification number, OP_{id} and a user password (shared among operators). There is a trusted setup phase at the utility site prior to any communication, when OP_{id}–password database is stored on HDs. In addition, key based hash functions (e.g., SHA-2), pseudorandom generator function, necessary crypto algorithms such as symmetric key algorithm (e.g., AES) and public-key encryption algorithm (e.g., RSA) are agreed upon and installed on both HD and TD. The installation and update configuration of functions/keys on TD are critical and out of the current scope. We use both *symmetric* and *public-key encryption algorithms* for protocol message communication over the wireless network, and *symmetric-key encryption algorithm* for telemetric data readings.

D. Attack Model

Since the whole communication system exists in an open environment, security barriers to prevent unauthorized access are very necessary. In this paper, we only consider cyber-security attacks. Physical attacks and security protections against them are out of scope of this paper. An attacker may try to get access of the devices by faking identities if the attacker gets the shared user password. Besides, since the network is wireless, attacker may eavesdrop on the communications and place man-in-the-middle attack on-site or a replay attack at later time. Even worse, attackers may get access to TD, break cryptographic keys information, and falsify telemetric data. A detailed security analysis is provided in section IV.

E. Setup and Assumptions

Utilities deal with a large number of TDs. Operators collect the measurements from a TD using HDs. Multiple operators may use the same HD at different days to collect telemetric data. On a particular day, operators ($OP_1, OP_2, ..., OP_i$) use handheld devices ($HD_1, HD_2, ..., HD_i$) to collect data from telemetric devices ($TD_1, TD_2, ..., TD_j$) at different locations ($L_1, L_2, ..., L_j$) at different times ($TS_1, TS_2, ..., TS_j$).

For maintaining confidentiality and authenticity of initial setup messages, public-private key pairs (PU_j-PR_j) are defined for each TD, and stored inside the HDs. However, since TDs are memory-constrained devices, instead of storing public keys of all HDs we generate on-the-fly symmetric keys using *PUFs* [3] attached with TDs to ensure a key agreement between HD and TD. This symmetric key is only used for securing initial protocol messages between them.

A *PUF* implements an on-chip physical function *puf*: $C \rightarrow R$ that takes an input challenge $Ch_i \in C$ and produces a response $Rs_i \in R$, where (C, R) is the set of all possible challenge-response pairs (CRPs). PUF relies on the intrinsic randomness during the integrated circuit fabrication process [14]. Therefore, CRPs cannot be cloned or reproduced exactly, not even by its original manufacturer, and is unique to each PUF [15]. We assume that the PUF system-on-chip (SoC) is integrated with each TD. During the trusted setup phase at the utility site, the utility constructs CRPs for the PUFs inside each TD, which is also stored into HDs' databases.

PUF can generate volatile cryptographic keys with low-cost [3] when a challenge is given. In practice, error correction codes (e.g., Reed-Solomon) are used to remove the noise from the PUF response and make it stable and identical. The output of the error correction unit (ECU) of length t is hashed down by the hash function H_1: $\{0, 1\}^t \rightarrow \{0, 1\}^m$ to a desired key KC of length m. KC is used for communication between HD and TD during the initial setup (detail in Section III). The key generation process using PUF is shown in Figure 2.

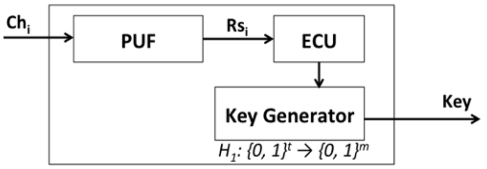


Fig. 2. Key generation using PUF

We assume that TD_j only stores its own private key (PR_j) and a shared secret with HDs in form of salt ($S_{cur,j} < 1$) in its firmware. On the other hand, HD_i has a list of public keys (PU) of all TDs in its memory in addition to all CRPs associated for all PUFs (in TDs). HD also stores a list of shared secrets, i.e., salts ($S_{cur,1,}$ $S_{cur,2}$, …, $S_{cur,j}$) in its firmware. In addition, a database of OP_{id}-password of all operators is stored in HD for human (operator) authentication. Both devices have the capability to execute AES, RSA, SHA-2 cryptographic algorithms and functions (defined in the following sections) to generate device passwords and one-time shared keys (P).

TABLE I. MATHEMATICAL NOTATIONS

Symbol	Definition
OP, HD, TD	System Principals (Operator, Handheld Device, Telemetric Device)
$E_{PUj}()$	Encrypt operation with Public Key of j^{th} TD
$D_{PRj}()$	Decrypt operation with Private Key of j^{th} TD
$[M]_{PR}$	Sign a message *M* with own private key
$E_P()/D_p()$	Encrypt/Decrypt with symmetric key, *P*
$E_{KC}()/D_{KC}()$	Encrypt/Decrypt with symmetric key *KC*
P	Session shared key of 256 bit
KC	Symmetric key generated by PUF
$p^/$	*k* bits of *P starting from index n*
k	Number of bits of *P* to verify
Ch_k^j	Challenge for PUF associated with j^{th} TD chosen from a set of *k* challenges
$S_{cur,j}$, $S_{prev,j}$	Salt (current and previous) at j^{th} TD
L	Location
TS	Time variant nonce
nonce	Random number
HD_{id}	Handheld Device id - 48bit MAC address
OP_{id}	Operator Identification number
ACK	Acknowledgement
ERR	Error message
TER	Terminate message
$f()$	Pseudorandom generator function
$Q()$	256-bit cryptographic hash function
$\|\|$	Append Operation

III. APPROACH

In this section, we present our password-changing protocol that provides robust authentication and secure communication. We divide our approach into three phases: **Phase 1** performs authentication of an operator (OP_k) to the handheld device (HD_i), **Phase 2** performs authentication between the handheld device (HD_i) and the telemetric device (TD_j), and **Phase 3** ensures secure communication between the handheld device (HD_i) and the telemetric device (TD_j).

The functionalities of the handheld device are built into the utility car. So, the OP authenticates into HD once (phase 1) when s/he starts driving for collecting data, and not at each pole. Moreover, operator authenticates his/her HD to each TD with a unique device password at each pole location to collect data readings (phase 2 and 3). Mathematical notations of the symbols are given in Table I.

A. Phase 1. In our approach, we consider the knowledge factor of the operator (e.g., password, PIN) since it is easier to use, convenient and less expensive to deploy than token-based or biometric methods. To authenticate, OP_k provides a valid unique user identification number (OP_{id}) and shared user password to HD_i. However, remote software robots may try to get access of TD by breaking into HD. To protect that, a CAPTCHA test [7] is introduced. HD generates a CAPTCHA using cyber-physical information (i.e., GPS location, temperature, and handheld device id HD_{id}), which is collected at the beginning of phase 1 using respective sensors. Fig. 3 formalizes the procedure of a robust authentication of OP in phase 1. Authentication of OP is important so that responsible OP can be identified in case of an insider attack. After OP authenticates, HD sends the login request message to TD (in the next phase).

OP_k	HD_i
	1.1 Generate a CAPTCHA
1.2. Enter Answer of CAPTCHA	
	1.3 Verify CAPTCHA Answer
2.1. Enter OP_{id} and Password	
	2.2 Verify OP_{id}, Password

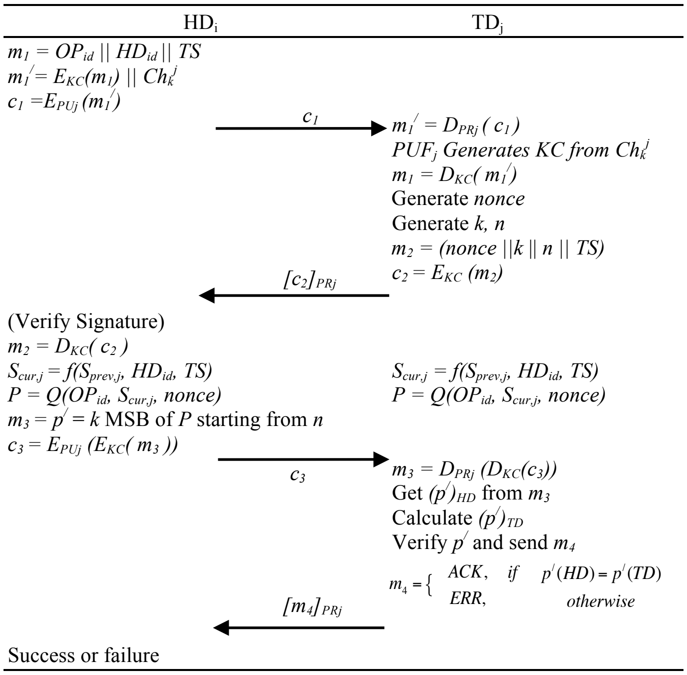
Fig. 3. Phase 1 - Authentication of operator

Phase 1 is associated with a timer or counter. When the counter expires (e.g., after few hours or visiting few different locations), the OP needs to perform re-authentication.

B. Phase 2. In this phase, HD authenticates itself to the TD and calculates the session-shared keys (to be used for transmitting telemetric data). As soon as the OP comes within the range of TD's wireless network, HD combines OP_{id}, HD_{id} (collected in phase 1) and time variant nonce TS in the form of a **login request message** m_1. HD then chooses a challenge-response pair, Ch_k^j-Rs_k^j from a set of k CRPs stored for j^{th} TD, and generates a key KC by hashing Rs_k^j (using hash function H_1:$\{0,1\}^t \rightarrow \{0,1\}^m$). HD encrypts message m_1 with KC and appends Ch_k^j so that TD can regenerate key KC from Ch_k^j using the PUF SoC (section IIE). Thus a volatile key, KC is agreed between TD and HD without storing additional keys in TD. The HD then encrypts again with the public key of TD and transmits the encrypted message (c_1 in Figure 4) to TD over the wireless network to initiate a conversation with the TD. Note that we do not require any clock synchronization between TD and HD.

When TD receives m_1, it extracts challenge Ch_k^j by decrypting c_1 using its own private key PR_j, generates key KC from PUF (using Ch_k^j), decrypts the rest of the message with KC and identifies OP_{id}, TS and HD_{id}. TD generates a random *nonce, k* and *n,* where both *k* and *n* are chosen from a range of numbers. A message m_2 is constructed by appending *nonce, k, n,* and extracted TS (from m_1) together. Then m_2 is transmitted to HD after encrypting with KC and signing with TD's private key to ensure the confidentiality and authenticity of the message (c_2 in Figure 4). Next, both devices (TD and HD) start calculating the same P using the equation: $P = Q\ (OP_{id}, S_{cur,j}, nonce)$. This P is a symmetric key used in the final phase for en/decrypting telemetric data. And $Q()$ is a 256-bit cryptographic hash function, e.g., SHA-2.

In function $Q()$, we use a salt value, $S_{cur,j}\ (<1)$ as an input, which is calculated using a pseudorandom generator function $f()$ with seed $[S_{prev,j} \,||\, HD_{id} \,||\, TS]$. At every session, a new $S_{cur,j}$ is calculated, which becomes $S_{prev,j}$ at the end of the session to be used for the next session. This way, value of *salt* changes for every session based on a secret value ($S_{prev,j}$) stored in the firmware of both TD and HD (installed beforehand). Hence, it is hard for the attacker to guess the value of $S_{cur,j}$.



HD_i
TD_j
$m_1 = OP_{id} \,||\, HD_{id} \,||\, TS$
$m_1^{/} = E_{KC}(m_1) \,||\, Ch_k^j$
$c_1 = E_{PUj}(m_1^{/})$
c_1
$m_1^{/} = D_{PRj}(c_1)$
PUF_j Generates KC from Ch_k^j
$m_1 = D_{KC}(m_1^{/})$
Generate *nonce*
Generate k, n
$m_2 = (nonce \,||k\,||\, n \,||\, TS)$
$c_2 = E_{KC}(m_2)$
$[c_2]_{PRj}$
(Verify Signature)
$m_2 = D_{KC}(c_2)$
$S_{cur,j} = f(S_{prev,j}, HD_{id}, TS)$
$P = Q(OP_{id}, S_{cur,j}, nonce)$
$S_{cur,j} = f(S_{prev,j}, HD_{id}, TS)$
$P = Q(OP_{id}, S_{cur,j}, nonce)$
$m_3 = p^{/} = k$ MSB of P starting from n
$c_3 = E_{PUj}(E_{KC}(m_3))$
c_3
$m_3 = D_{PRj}(D_{KC}(c_3))$
Get $(p^{/})_{HD}$ from m_3
Calculate $(p^{/})_{TD}$
Verify $p^{/}$ and send m_4
$m_4 = \begin{cases} ACK, & \text{if} \quad p^{/}(HD) = p^{/}(TD) \\ ERR, & \text{otherwise} \end{cases}$
$[m_4]_{PRj}$
Success or failure

Fig. 4. Phase 2 - Authentication of *HD* to *TD* and Shared Key Generation

Note that $S_{cur,j}$ values vary across TDs. Once $S_{cur,j}$ is assigned as $S_{prev,j}$ for future computations, the updated $S_{prev,j}$ needs to be disseminated to other HDs before they access the same TD. The synchronization of $S_{prev,j}$ across the HDs is done at the end-of the day at the utilities. Since one TD is accessed maximum once a day, synchronization of $S_{prev,j}$ at the utilities does not require any behavioral changes in the measurement.

In our password changing protocol, only k bits from index n of shared-symmetric key P are used as device password ($p^{/}$) for the authentication of HD to TD (different from the OP's shared password entered in phase 1). HD picks $p^{/}$ (message m_3 in fig. 4), encrypts it with KC and public key of TD and transmits the encrypted message c_3 to the TD as the computed response. Upon receiving c_3, TD decrypts it and extracts $p^{/}$ calculated by HD. The TD then validates received $p^{/}$ with the self-computed $p^{/}$. If the received and local $p^{/}$ 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 message m_4 is also sent after signing it with the private key, so that the HD knows that this message comes from the legitimate TD. $S_{prev,j}$ is updated only when an authentication is successful. The authenticity and confidentiality of the messages is maintained by using both KC and the private key (PR_i) of TD. Note that KC is not used as the symmetric key for telemetric data encryption in phase 3, since KC repeats for the same input Ch_i, which invalidates the notion of one-time password and key generation. Therefore, one-time key P is generated in this phase.

C. Phase 3. In this phase, secure delivery of the telemetric data is ensured. Both devices use the *256* bits P derived in phase 2 as the symmetric key. The TD_j reads the telemetric measurements from memory, encrypts the data with P, signs and sends it to HD_i over the wireless network. Upon receiving the data HD_i stores them in secure database. Finally, they conclude when TD_j sends a signed termination message to ensure that the session is terminated. Fig. 5 shows the details.

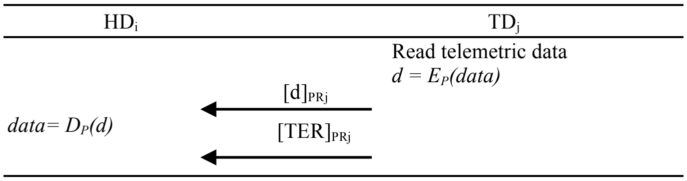


Fig. 5. Phase 3 - Communication between two devices

IV. Security Analysis

In this section, we analyze the security of SCAPACH against various cyber-attacks that are considered important to address in literature [9][13].

Replay Attack: In SCAPACH, the device password keeps changing across each data collection session. Therefore, even if an attacker eavesdrops the flying message (c_3), she cannot use it for future sessions to authenticate. Moreover, to protect against replay of c_1, an alternative of our protocol can be formulated. While constructing $m_1^{/}$ (as shown in Figure 4), HD can encrypt m_1 separately with secret $S_{cur,j}$ and KC, and send both of them in $m_1^{/}$. Since $S_{cur,j}$ is different for each session, TD can check whether c_1 is intended for current session or not by verifying both m_1. Thus our protocol thwarts replay attack. This altrenative also thwarts **Denial of Service (DoS)** attacks. However, this alternative introduces another level of decryption and hence, there is a tradeoff between the computational cost and security against the DoS attack.

Perfect Forward Secrecy: The forward secrecy property ensures that the conversation an adversary recorded remains secret if one of the private keys is compromised in the future. In our protocol, even if an intruder gets access to private key of TD, she cannot derive the messages exchanged. It is because, m_1, m_2, m_3 are encrypted using KC, which changes depending on input challenges. Therefore, even if the private key is compromised, attacker cannot derive P; hence SCAPACH maintains perfect forward secrecy.

MITM Attack: A man-in-the-middle (MITM) attack requires an attacker to fool both sides of a legitimate conversation [5]. This is not possible in our protocol since a key agreement needs to be established between HD and TD (more discussed below).

Masquerade of Telemetric Device: All messages sent from HD are encrypted with the public key of TD. To protect the masquerade of TD, HD sends time variant nonce (*TS*) that TD need to send back in m_2. HD makes sure that it is talking to a legitimate TD by verifying the received *TS* in c_2. Because, attacker does not have the private key of TD and hence cannot decrypt and reveal correct *TS* from c_1. Also, TD signs the messages with its private key PR_j, which also ensures HD that it is communicating with a legitimate TD.

Masquerade of Handheld Device: An intruder can generate a garbled c_1 and send it to TD to pretend like HD. The TD extracts *Ch* from received message and the associated PUF generates the key *KC* (using garbled *Ch*), which both parties need to use for further communication. However, according to the property of PUF, an attacker can never produce correct *Rs* from a given *Ch* [14]. PUFs can only be broken by numerical modeling attacks if the attacker knows a set of CRPs of a PUF [15]. However, no CRP is revealed during any communication in our protocol. So, the attacker can never derive correct *KC* and hence cannot decrypt c_2. Moreover, TD sends the *nonce* that is supposed to be used as an input to calculate *P* only for that session. Therefore, the intruder can never compute a valid *P* and hence cannot pretend to be an authorized HD.

V. Implementation and Evaluation

To validate the SCAPACH protocol, we use two laptops as HD and TD with Intel Core 2 Duo 2.26GHz processor and 2GB read-only memory. The prototype is implemented in Java so that it can be easily ported into mobile phone-like devices. The communication between laptops uses wifi 802.11n wireless network. RSA is used as public key encryption. However, AES is used as symmetric key algorithm to encrypt telemetric readings, since it is faster for larger size of data.

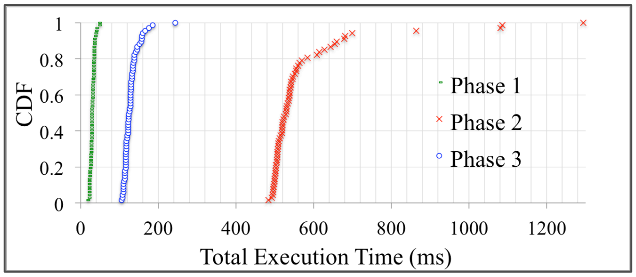


Fig. 6. Execution time of SCAPATH over 100 executions

To compute the performance of SCAPACH, we measure its total execution time, which is 730ms on average. Fig. 6 shows the CDF of execution times of SCAPACH in three phases over 100 executions. Since, the authentication process of operator in Phase 1 is done locally at HD and the operator-side delay is considered negligible, the execution in Phase 1 is very small (average 26.7ms). Phase 2 takes the highest time due to the repeated communication between TD and HD; however, the execution time is less than 600ms in most of the cases (80% cases in Fig. 6). The execution time in Phase 3 is about 150ms on average. Note that we do not consider the computational time of *PUF* here since it is a separate SoC and can generate keys in parallel with TD. The network communication delay is about 15-20ms. We also measure the execution time of different processes (e.g., encryption, computation of *P*, etc.) inside each phase. We find that the RSA encryption-decryption time is the main contributor to the execution time in phase 2 and phase 3. Our implementation results indicate that SCAPACH works efficiently.

VI. Conclusion

In this paper, we highlight one of the realistic authentication problems in the smart grid critical infrastructure. We propose a secure authentication and data transmission protocol for collecting telemetric data from the pole devices. Our password-changing framework, SCAPACH**,** creates short-lived passwords and shared keys based on physical characteristics (such as per-pole device locality, data collection timestamp and per-driver identification) and changeable secret; and ensures secure data collection considering the resource-bound limitations of the telemetric devices. The protocol is fast and secured against different security attacks in this domain.

Acknowledgment

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

References

[1] S. M. Bellovin et al., “Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks,” In *Proc. of SP,* 1992.

[2] S. Sridhar et al., "Cyber–Physical System Security for the Electric Power Grid," *Proc. of the IEEE*, vol.100, no.1, pp.210, 224, 2012.

[3] G. E. Suh et al.,“Physical unclonable functions for device authentication and secret key generation,” In *Proc. of DAC*, pp. 9–14, 2007.

[4] M. Steiner et al., “Refinement and extension of encrypted key exchange,” ACM SIGOPS, 1995.

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

[6] S. M. Bellovin et al., “Augmented Encrypted Key Exchange: a Password Based Protocol Secure Against Dictionary Attacks and Password File Compromise,” *AT&T Bell Laboratories,* 1994.

[7] L. von Ahn et al., “Telling Humans and Computers Apart (Automatically) or How Lazy Cryptographers do AI,” *Communications of the ACM,* 2004.

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

[9] M.M. Fouda et al., "A Lightweight Message Authentication Scheme for Smart Grid Communications," *in Proc. of IEEE Smart Grid*, vol.2, no.4, pp.675, 685, 2011.

[10] Personal communication with Ameren, *TCIPG industry Workshop*, 2012, Urbana, IL.

[11] A. Shamir, “Identity-based Cryptosystems and Signature Schemes,” in *Proc. of CRYPTO,* 1985.

[12] S.H.M Kwok et al., “Zero-Configuration Identity-Based Signcryption Scheme for Smart Grid,” in *Proc. of* SmartGridComm, 2010.

[13] B. Y. Mo et al., “Cyber-physical security of a smart grid infrastructure,” *Proc. IEEE*, vol.100, no. 1, pp. 195–209, Jan. 2012.

[14] M. Nabeel et al., “Authentication and Key Management for Advanced Metering Infrastructures Utilizing Physically Unclonable Functions,” In *Proc. of IEEE SmartGridComm*, 2012.

[15] U. Rührmair et al., “Modelling Attacks on Physical Unclonable Functions,” In *Proc. of CCS ’10*, pp. 237-249, 2010.