A Secured Offline Authentication Approach for Industrial Mobile Robots

Sarah Haas,[1] Andreas Wallner,[1] Ronald Toegl,[1] Thomas Ulz[2] and Christian Steger[2]

***Abstract*— Mobile robots are used to replace conveyors in production facilities as they provide more flexibility and are easier to install or replace. These robots suffer from higher safety risks than conveyors as they move freely, necessitating extended security needs. A major point is the need for authentication to prevent unauthorized persons from manipulating a robot's software or configuration. Traditional username and password schemes are unwildy and insufficient for industrial mobile robots as administration and maintenance do not scale well. We propose the use of one-time passwords for authentication on robots based on a shared secret and a counter. The authentication mechanism is further supported by secure elements to allow secured storage of the key and secured password derivation. We also provide a threat analysis for the proposed methods.**

I. Introduction

The increased automation in production facilities entailed the use of Industrial Automation Robots (IARs) to decrease the number of workers and increase the production time as IARs can work 24 hours a day. *Smart Factories* [1] extend the use of robots in production facilities beyond IARs and introduce Industrial Mobile Robots (IMRs) to replace conveyors and move production material between IARs or manipulate products [2]. A main advantage of IMRs over conveyors is their flexibility and the fact they can easily be replaced by another IMR. However, both IMRs and IARs pose high safety risks as they move in the same environment as humans [3].

IMRs could be described as computer systems with wheels, sensors and actuators. As such it is imminent that safety is directly connected to the security of the used computer system. To provide security, access control and rights of users have to be limited to prevent unauthorized users from manipulating an IMR's software or settings. Therefore, an authentication mechanism to identify authorized users can be understood as essential for IMRs.

When accessing an IMR locally, username and password would be a possible option. Such access control becomes unwildy very fast considering the many IMRs and their administration. Storage of all username and password tuples is needed on all IMRs if one does not want to rely on an authentication server which would possibly be at a remote location. Public key cryptography is a solution for remote authentication, for local authentication this also poses administrative problems. Due to this large number of problems, we propose the use of one-time password, which are passwords that can only be used once and do not rely on a specific user. This eliminates problems with shoulder surfing, password changes and still allows for fine grained control and time limited access. Authentication mechanisms supported by hardware security should be used to protect the authentication credentials from physical attacks [4].

The methods used for authentication should allow for offline processing, not needing another machine, server or internet access to work. This would be especially a problem for e.g. rented IMRs where a working internet connection to the owners infrastructure would be needed. The need for an online connection would also be problematic in the case of hardware failures.

As classical passwords and asymmetric methods are infeasible and online authentication is prone to connectivity problems, we propose an authentication mechanism for the direct access to IMRs in production facilities by using One-Time Passwords (OTP), secure elements and smart cards. The authentication mechanism provides two different authentication modes to provide more flexibility. Both authentication modes are utilized for the purpose that the passwords do not rely on a persons credentials but is derived from the identity of a specific IMR or group of IMRs that is accessed. To the best knowledge of the authors, no such approach was previously proposed. Therefore, the contributions of this paper are

1) a feasible, secured method for offline authentication on mobile robots for authorized personnel in an industrial context, and
2) an extension of this method to allow access to groups of robots, and
3) a threat analysis for the proposed methods.

The remainder of the paper is structured as follows. In Section II the related work and background regarding one-time passwords, smart cards for authentication, authentication on robots and a possible key derivation function are given. Section III contains the detailed description of the authentication approach, the recovery behavior and the two possible authentication modes. A threat analysis to show the security of the approach is described in Section IV. Finally, Section V contains the conclusion.

II. Related Work and Background

A. Authentication on Robots

To the best knowledge of the authors, the only existing offline authentication algorithms for users on mobile robots

[1]Sarah Haas, Andreas Wallner and Ronald Toegl are with Development Center Graz, Infineon Technologies Austria AG sarah.haas, andreas.wallner, ronald.toegl@infineon.com

[2]Thomas Ulz and Christian Steger are with Insitute for Technical Informatics, Graz University of Technology, Austria thomas.ulz, steger@tugraz.at

are the ones by Kim et al. [5], [6]. Most other approaches address authentication of robots on other robots such as [7] or [8]. Kim et al. [5], [6] proposed two approaches for robot authentication mechanisms using biometrics to recognize human faces, body height, and clothes color. To authenticate a user, the robot tries to match the detected face, body height and clothes to an already known human. The evaluation results showed that humans were correctly authenticated even if just the body height was detected. However, the percentage of correctly detected humans decreased when only the body height was available. Kim's approaches require that the data of each authorized human need to be stored on the robot, which results in a huge administration effort similar to a username-password-based approach. Furthermore, the approaches perform image processing that requires a large computational effort.

B. One-Time Passwords (OTP)

One-time passwords are password schemes where one password becomes invalid after its use and were introduced by Lamport [9] to prevent eavesdroppers from revealing a user's plain password and later, as a countermeasure against replay attacks that try to capture the (hashed) login credentials of users and use them to access the system [10]. OTPs require the same non-invertible cryptographic hash function on the client and the host.

In 2005, M'Raihi et al. [11] proposed *HOTP* (HMAC-based OTP), another approach for OTPs that expire after the first usage. The authors introduced a counter that is added to the hashing function in combination with the secret key. The counter must be synchronized with a trusted entity such as a server to enable the verification of the OTP. Each time an OTP is computed, the counter is incremented by a specific amount known by the client and server. It is possible that the server's and client's counter are not synchronized anymore, which is detected by the server. The server then calculates several OTPs by increasing the counter and matching them to the received OTP from the client. If synchronization fails and the client reached the limit of authentication attempts, the client is locked out.

M'Raihi et al. [12] also proposed *TOTP* (Time-based OTP) which is based on HOTP and uses OTPs that expire after a specific amount of time. The main difference is that HOTP is an event-based algorithm where the counter is the moving factor and the password is changed whenever a new password is required. TOTP, on the other hand, uses time as a moving factor that is independent of any event. This means, that with HOTP OTPs are only generated when the user triggers the event of requesting a new OPT, whereas with TOTP OTPs are generated every time a specific amount of time elapsed even if a user did not require a new OTP within this amount of time.

C. Smart Card-based Authentication

Song et al. [13] proposed an OTP-based authentication approach using a smart card. First, the user sends username and password to a server that verifies the user's identity and afterwards the server sends a smart card to the user containing a user ID, a shared value, a hashing function and a symmetric encryption function. Later, when trying to log in, both server and smart card verify exchanged messages and grant access to the server if valid.

Vaidya et al. [14] propose a similar approach using smart cards to authenticate on home gateways rather than on servers. In their approach, the first step is to register the user at an Integrated Authentication Server (IAS) with his username and password through the Internet. In contrast to the approach by Song et al. [13], the authors did not store the user's username and password on the server to verify the identity but a hash that was derived from a user's provided data. The derived hash was then, among other things, compared to the one sent by the smart card for authentication. The second step is to connect to the IAS using the smart card to create an OTP that grants the user access to the home gateway. The OTP does not expire after the first use but has a lifetime that is influenced by the time and the number of authentication attempts and is set by the IAS. The authors further show that their approach is not prone to most known attacks on smart card-based authentication schemes such as stolen-verifier or forward secrecy.

Xu et al. [15] show that many approaches are prone to attacks due to the fact that values stored in stolen or lost smart cards can be revealed when applying a power analysis. Depending on the authentication approach, the values from the smart card can be enough to log in to the system without knowing the user's password. The authors also propose an approach that prevents the possibility of logging in to a system with a stolen smart card by using the Computational Diffie-Hellman Assumption.

D. HKDF

The HMAC-based Extract-and-Expand Key Derivation Function (HKDF) [16] is a key derivation function (KDF) using Hash-based Message Authentication Codes (HMAC) [17]. A cryptographic hash function is a mathematic function that maps data of arbitrary size to a bit sequence of fixed size and is infeasible to invert. Message authentication codes (MAC) are used to verify the integrity of a message to ensure that the message was not modified during transit. A hash-based MAC (HMAC) is a specific MAC using a cryptographic hash function and a secret cryptographic key to provide message integrity. A Key Derivation Function (KDF) is used to derive a strong cryptographic key from some initial keying material. HKDF is a KDF that uses an HMAC to compute a strong cryptographic key and is split into an extract and an expand step. In the extract step, an HMAC is used to compute a pseudorandom key from a secret master key and an optional salt. In the expand step, the output key is computed from the pseudorandom key and some optional message again using a cryptographic hash function. HKDF can be used to derive keys of different length, while for the proposed use-case only the expand step is used (generation of a random key using data known on multiple devices). The HKDF expand step will be written as

TABLE I
AUTHENTICATION FLOW OF HOTP-LIKE METHOD

	MD		IMR
1 :	$deriv_g \leftarrow \mathrm{HKDF}(K_M, cnt_i)$		
2 :	$cnt_g \leftarrow cnt_g + 1$		
3 :	$pw_g \leftarrow left(deriv, n)$		
4 :		$\xrightarrow{pw_g}$	
5 :			**for** $n = 0..9$ **do**
6 :			$deriv_i \leftarrow \mathrm{HKDF}(K_M, cnt)$
7 :			$pw_i \leftarrow left(deriv_i, n)$
8 :			**if** $pw_i = pw_g$
			$cnt_i \leftarrow cnt_i + n$
			$return$ success
9 :			$errorCnt \leftarrow errorCnt + 1$
10 :			$return$ failure

$\mathrm{HKDF}(key, data)$, where key is the base key and data is the information used for derivation.

E. Secure Element

A secure element is a hardware device that is intended to host security critical applications and store cryptographic or confidential data. It is tamper resistant and provides cryptographic functionality and memory that is protected against physical attacks. In contrast to normal microcontrollers or memory devices where it is generally easy to extract stored content, influence or spy on calculations, these devices are hardened so that considerable effort is necessary to achieve similar results. Examples for such devices would be chipcards used e.g. for banking applications, trusted platform modules [18] or Public-Key Cryptography Standard (PKCS#11) [19] implementations on hardware tokens.

III. SINGLE AND GROUP ROBOT AUTHENTICATION

The importance of IMRs has increased in the last few years due to their abilities to move freely, to cooperate when manufacturing products or to manipulate products. As these IMRs need maintenance to work properly, the authentication scheme proposed in this paper allows to check that only authorized personnel can access an IMR's interfaces. As many different users might be authorized to access an IMR, OTPs are used for authentication. Password and username are infeasible in this scenario as the credentials for each user would have to be stored and maintained on each IMR separately. The use of OTPs enables the authentication to be independent from a specific user and still provide secured authentication on the IMRs.

The approaches proposed here are based on *HOTP* [11] combined with hardware security elements and modified to use a mobile device instead of tokens. The authentication approach consists of the following entities:

- **User**: Any employee that needs access to IMRs.
- **Backend:** Infrastructure at the maintainer of the IMRs. Generates and stores the master key K_M as well as the last known counters for each IMR. Also, derives time-limited keys.
- **Mobile Device (MD):** An MD that will be used for OTP generation.
- **IMR:** Is pre-personalized with the master key K_M at production or delivery. The IMR uses a secure element for key storage and OTP processing.

The following subsections describe the authentication mode for single robot authentication. The difference to the mode for group robot authentication will be described afterward in Subsection III-D.

A. HOTP-based authentication

The HOTP algorithm [11] can be modified to use an MD instead of tokens and compute the OTP directly on the MD. Initially, the user needs to copy the master key K_M and the current counter value for the IMR from the backend to the used MD at the IMR's manufacturing company. An important precondition is that the MD used in this approach needs to be a secured smart card to protect the master key from being revealed if the MD is lost or stolen. The password verification is shown in Table I.

The user enters the IMR's ID number into the MD (manually or e.g. via barcode or NFC). The MD then calculates an OTP from K_M and the stored counter value for the IMR in question (1). The counter value stored by the MD is then increased (2). The OTP is truncated to the desired security level, converted into an easily typable representation and presented to the user (3). The user enters the OTP on a login screen on the IMR (4).

The IMR derives n OTPs starting from its current counter value where n defines an interval in which one value needs to be used to constitute a valid OTP (5). The OTP entered by the user is checked against those n generated IMR OTPs

and access is granted if the entered OTP matches one of these (8). The incremented counter value of the matching OTP is then stored. If no matching OTP is found on the IMR, access is denied, an error counter is incremented and the IMR locked after a certain number of attempts (9). After return of the user to his home base, the counter values in the backend are updated. There they can be used for additional checks or audits.

In the ideal case, the counter values of MD and IMR match exactly. This would not be the case in two situations: First, the counter value in the MD can be higher than in the IMR if e.g. an OTP was generated but never used. For this reason a window of possible OTPs is calculated by the IMR on verification. Second, if the IMR has a higher count than the MD, we consider the information in the MD to be out of date and deny access. A higher counter value in the IMR indicates that the MD was not synchronized or used for authentication for a long time. Therefore, it should not be allowed to grant access to the IMR anymore.

This basic scheme shows the following properties:

- Breaking access to one IMR and extracting K_M does not automatically grant access to all IMRs (because the counters of those would still be unknown).
- It is hard to support multiple users without online connection due to diverging counter values.
- If the MD is lost or stolen, the credentials stored on it only become invalid by the continuous logins using other devices. In that case, the counter values in the IMR is higher than on the MD and therefore make the credentials invalid.

B. Enhanced authentication

The approach described above can be extended by adding an additional derivation step in the backend to allow the usage of an insecure MD such as a smart phone. The use of an insecure MD simplifies the authentication as no additional smart cards are necessary. However, a secured smart card could still be used to provide further security. The master key K_M is not stored on the MD, which reduces the risk when the device is lost or stolen.

Before setting up the MD, a time-limited intermediate key $K_I \leftarrow \mathrm{HKDF}(K_M, expDate)$ is derived from K_M and an expiration date $expDate$. Both, the intermediate key and the expiration date, are stored on the MD.

As shown in Table II, when authenticating, the expiration date is first transmitted to the IMR (1) to check if the credentials are still valid (2). If so, it replies with the current authentication counter (3). The MD then uses the same procedure as before to calculate a valid OTP.

For verification, the IMR first calculates the K_I that the MD must be using from the expiration date received (8). The IMR then uses this K_I to calculate the OTP using its current counter value (9). If the IMR's calculated OTP matches the one received, access is granted (11). Otherwise, access is denied. After a successful authentication attempt, the counter is incremented to invalidate the just entered password after use. If the password is found to be invalid, an error counter is increased to prevent brute-force attacks and lock the IMR completely after too many incorrect tries (12).

After return of the user to his home base, the counter values could be copied to the backend for possible later usage for e.g. predictive maintenance.

As a communication medium we recommend to use close range communication like NFC. For legacy devices data transfer can be achieved by displaying the information and letting the user type it into the devices. In this case it would be appropriate to shorten the calculated password and convert it to an easily typeable representation. The length the key can be shortened to depends on the needed security level, but can generally be short since only OTPs are used and only brute-force protection is needed. In any case, password protection or similar is needed on the MD to limit the impact of lost/stolen devices.

The scheme shows the following properties:

- K_M is not stored on the MD so that it does not need to be secure anymore and the K_I becomes invalid over time.
- Multiple users are by definition supported as the counter does not rely on a single user.
- Breaking an MD only grants access until the stored K_I becomes outdated and invalid.

C. Recovery Behavior

The number of authentication attempts that can fail consecutively is restricted to 10 to limit attacks on the expiration date and counter. If the number of authentication attempts is not limited an adversary could, for example, try to brute-force the expiration date. After 10 failed authentication attempts the authentication is locked and can be reactivated with a reactivation key K_R. The reactivation key can be applied to the robot via a secured smart card. The reactivation key is not derived but unrelated and fixed and should not sent to the MD by default. The reactivation key only activates the authentication but does not authenticate the user. This mechanism is used to avoid the possibility of an adversary breaking the reactivation key and directly entering the system.

D. Authentication Modes

The proposed authentication approach describes the first of the two authentication modes. In the first mode, single-robot authentication, each IMR holds its own master key and derives OTPs from this master key and some input data. To make this authentication approach scalable for large factories with a huge number of IMRs, a second authentication mode can be considered. In the second mode, group-robot authentication, a group of IMRs holds the same master key and derives the OTPs from this shared key. This approach is applicable when several IMRs should be maintained at the same time. The two authentication modes can be used in parallel, meaning that the IMR holds both, an individual master key and a group key. Depending on the necessary action the better suited mode can be used for authentication.

TABLE II

AUTHENTICATION FLOW OF THE ENHANCED AUTHENTICATION APPROACH

	MD		**IMR**
1 :		$expDate \longrightarrow$	
2 :			**if** $expDate < now$ $return$ failure
3 :		$\longleftarrow cnt$	
4 :	$deriv_g \leftarrow \mathrm{HKDF}(K_I, cnt)$		
5 :	$pw_g \leftarrow left(deriv, n)$		
6 :	$cnt_g \leftarrow cnt_g + 1$		
7 :		$pw_g \longrightarrow$	
8 :			$K_{I_i} \leftarrow \mathrm{HKDF}(K_M, expDate)$
9 :			$deriv_i \leftarrow \mathrm{HKDF}(K_I, cnt)$
10 :			$pw_i \leftarrow left(deriv_i, n)$

11 :			**if** $pw_i = pw_g$
			  $cnt_i \leftarrow cnt_i + 1$
			  $return$ success

12 :			$errorCnt \leftarrow errorCnt + 1$
13 :			$return$ failure

IV. THREAT ANALYSIS

A threat analysis [20] was performed to show the achieved security level and also to emphasize the security features of the proposed approach. The analysis was done for the enhanced approach as the threat analysis for the HOTP-like approach was already done in [11]. The threat analysis lists all **Entities (E)**, **Assets (A)** that need protection, possible **Threats (T)**, necessary **Assumptions (As)** as well as **Countermeasures (C)** and **Residual Risks (R)**.

The entities involved in the authentication process and assumptions regarding their trustworthiness are:

- (E1) User: (As1) not trustworthy, possible adversary
- (E2) MD: (As2) trustworthy
- (E3) IMR: (As3) trustworthy
- (E4) Secure Element: (As4) trustworthy
- (E5) Malicious Adversary: (As5) not trustworthy, able to perform attacks on equipment

Before discussing the threats, several assumptions regarding the authentication approach need to be stated:

- (As6) The IMR uses a hardware secure element to store the key, generate the OTPs and check the OTPs provided by the MD.
- (As7) The master key K_M is already stored on the IMR and the backend.
- (As8) The counter cnt was initially synchronized between the backend and the IMR.
- (As9) The backend is assumed to be sufficiently secured against relevant of attacks and will therefore not be addressed in the threat analysis.

Using the entities, the assets that need to be protected can be listed:

- (A1) Keys: master key K_M and intermediate keys K_I need to be protected. Loss of keys would enable an adversary to reveal the counter and access an IMR.
- (A2) Counter: Counter should to be protected. Loss of counter might enable easier attack.
- (A3) Interface on IMR: Maintenance interface needs to be protected from unauthorized usage.

For each threat, the assets and entities as well as the possible countermeasures and residual risks are listed. The residual risks are threats that cannot be mitigated by the proposed approach.

- (T1) Intentional and unintentional backdoors in secure element on IMR.
  Entities: (E3), (E4), Assets: (A1), (A2)
  (C1) Secure elements are certified for specific security level. Certificate proves that no backdoors exist.
- (T2) Wrongly implemented/weak cryptography on secure element.
  Entities: (E3), (E4), Assets: (A1), (A2)
  (C2) Secure elements are certified for specific security level. Certificate proves correctness of cryptography implementation and strong algorithms.
- (T3) Loss/theft of MD.
  Entities: (E1), (E2), (E5), Assets: (A1), (A3)
  (C3) By owning the intermediate key it is not possible to reveal the master key due to a strong cryptographic derivation algorithm.
  (C4) Revealing the expiration date from the intermedi-

ate key is not possible due to a strong cryptographic derivation algorithm.
(C5) Limited authentication attempts shut down brute-force attacks on the expiration date.
(R1) With shoulder surfing attacks it is possible to steal the expiration date for a user by spying over the user's shoulder. The MD must also be stolen to perform an authentication. Physical access to MD and IMR are necessary.

- (T4) Manipulation of counter on MD.
  Entities: (E1), (E2), (E3), (E4), (E5), Assets: (A3)
  (C6) Wrong counter generates invalid OTP which is not accepted by IMR.
- (T5) Manipulation of OTP on MD.
  Entities: (E1), (E2), (E3), (E4), (E5), Assets: (A3)
  (C7) Invalid OTP not accepted by IMR.
- (T6) Using same OTP several times.
  Entities: (E1), (E2), (E3), (E4), (E5), Assets: (A3)
  (C8) OTP becomes invalid after first usage.
- (T7) Physical attack on IMR.
  Entities: (E1), (E3), (E4), (E5), Assets: (A1), (A2), (A3)
  (C9) Secure element provides tamper resistance, extracting master key is not feasible for adversaries.
- (T8) DoS attack on IMR.
  Entities: (E1), (E3), (E5), Assets: (A3)
  (C10) Limited number of authentication attempts mitigate DoS attacks. Furthermore, authentication is performed on secure element, which does not affect IMR's functionality.
- (T9) Remote attack on authentication.
  Entities: (E1), (E2), (E3), (E5), Assets: (A3)
  (C11) Short communication range between MD and IMR limits possibilities for remote attacks such as MITM as physical access would be necessary.

The threat analysis is not exhaustive and list the most crucial threats identified by the authors. However, the analysis shows that only one residual risk remains when using the proposed approach.

V. Conclusion

In this paper, two offline authentication approaches for users on industrial mobile robots were proposed to overcome the issues with username/password based authentication schemes and connectivity problems of online approaches. The basic approach was extended to be able to use any untrustworthy mobile device to grant access to the robot by using derived time-limited keys. The threat analysis has shown that the approach provides a high level of security due to the fact that only one residual risk remains.

Acknowledgment

A part of the work has been performed in the project Power Semiconductor and Electronics Manufacturing 4.0 - (Semi40), under grant agreement No 962466. The project is cofunded by grants from Austria, Germany, Italy, France, Portugal and - Electronic Component Systems for European Leadership Joint Undertaking (ECSEL JU).

References

[1] D. Zuehlke, "SmartFactory–Towards a factory-of-things," *Annual Reviews in Control*, vol. 34, no. 1, pp. 129–138, 2010.
[2] S. Schneider, F. Hegger, N. Hochgeschwender, R. Dwiputra, A. Moriarty, J. Berghofer, and G. K. Kraetzschmar, "Design and development of a benchmarking testbed for the Factory of the Future," in *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE, 2015, pp. 1–7.
[3] I. Ranatunga, S. Cremer, F. L. Lewis and D. O. Popa, "Neuroadaptive control for safe robots in human environments: A case study," in *Automation Science and Engineering (CASE), 2015 IEEE International Conference on*. IEEE, 2015, pp. 322–327.
[4] S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, "Security in Embedded Systems: Design Challenges," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 3, pp. 461–491, 2004.
[5] D.-H. Kim, J. Lee, H.-S. Yoon, H.-J. Kim, Y. Cho, and E.-Y. Cha, "A vision-based user authentication system in robot environments by using semi-biometrics and tracking," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 1812–1817.
[6] D. Kim, J. Lee, H.-S. Yoon, and E.-Y. Cha, "A non-cooperative user authentication system in robot environments," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 804–811, 2007.
[7] W. Adi, "Mechatronic security and robot authentication," in *2009 International Symposium on Bio-inspired, Learning, and Intelligent Systems for Security, BLISS 2009*, 2009, pp. 77–82.
[8] M. L. Gavrilova and R. V. Yampolskiy, "State-of-the-art in robot authentication," *IEEE Robotics and Automation Magazine*, vol. 17, no. 4, pp. 23–24, 2010.
[9] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.
[10] N. Haller, "The S/KEY One-Time Password System," 1995.
[11] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, "HOTP: An HMAC-Based One-Time Password Algorithm," Tech. Rep., 2005.
[12] D. M'Raihi, S. Machani, M. Pei, and J. Rydell, "TOTP: Time-Based One-Time Password Algorithm," *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 1689–1699, 2013.
[13] R. Song, "Advanced smart card based password authentication protocol," *Computer Standards & Interfaces*, vol. 32, no. 5, pp. 321–325, 2010.
[14] B. Vaidya, J. H. Park, S.-S. Yeo, and J. J. P. C. Rodrigues, "Robust one-time password authentication scheme using smart card for home network environment," *Computer Communications*, vol. 34, no. 3, pp. 326–336, 2011.
[15] J. Xu, W.-T. Zhu, and D.-G. Feng, "An improved smart card based password authentication scheme with provable security," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 723–728, 2009.
[16] H. Krawczyk, "Cryptographic extraction and key derivation: The HKDF scheme," in *Annual Cryptology Conference*. Springer, 2010, pp. 631–648.
[17] H. Krawczyk, M. Bellare, and R. Canetti, "RFC2104 - HMAC: Keyed-hashing for message authentication," Tech. Rep., 1997.
[18] "ISO/IEC 11889-1 Trusted platform module library - Part 1: Architecture," 8 2015.
[19] OASIS Standard, "PKCS #11 Cryptographic Token Interface Base Specification Version 2.40," 2015.
[20] S. Myagmar, A. J. Lee, and W. Yurcik, "Threat modeling as a basis for security requirements," in *Symposium on requirements engineering for information security (SREIS)*, vol. 2005. Citeseer, 2005, pp. 1–8.