A Universally Composable Key Exchange Framework for Operational Technology Communication Protocols

Abubakar Sadiq Sani*, Elisa Bertino†, Dong Yuan‡, and Zhao Yang Dong§
* School of Computing and Mathematical Sciences, University of Greenwich, London, United Kingdom
Email: s.sani@greenwich.ac.uk
† Department of Computer Science, Purdue University, West Lafayette, Indiana, USA
Email: bertino@purdue.edu
‡ School of Electrical and Information Engineering, The University of Sydney, Sydney, Australia
Email: dong.yuan@sydney.edu.au
§ School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore
Email: zy.dong@ntu.edu.sg

***Abstract*—Many real-world communication protocols in Operational Technology (OT) are prone to cyber attacks due to the absence of dedicated key exchange and inability to satisfy all key exchange properties such as mutual authentication, key secrecy, and key confirmation. Besides these deficiencies, the security of communications in the OT is severely lacking at least one of the communication security properties such as data integrity, data confidentiality, and data availability. In this paper, we propose to enhance OT communication protocols via a secure key exchange framework for satisfying the key exchange and communication security properties based on universal composability, which provides modular design and analysis of cryptographic protocols in the presence of an adversary. Our framework comprises an ideal crypto-ops functionality (F_{cl}) for modelling and satisfying the key exchange and communication security properties. We analyse the security of our framework using the Automated Validation of Internet of Security Protocols and Applications (AVISPA) tool. Furthermore, we illustrate the usefulness of our framework by fixing one of the most widely used real-world OT communication protocols, namely WirelessHART.**

***Index Terms*—operational technology, key exchange, communication security, universal composability, communication protocols**

I. Introduction

Key exchange is one of the most widely adopted cryptographic protocols in many secure communication protocols. The key properties for key exchange are mutual authentication, key confirmation, and key secrecy which guarantee a secret key is shared between intended devices or users, the devices have the same secret key, and no other device knows about the secret key, respectively [1]. The general properties for communication security are data integrity, data confidentiality, and data availability. Many communication protocols such as Wireless Highway Addressable Remote Transducer (WirelessHART) protocol [2] use secret session keys to secure data exchange in Operational Technology (OT), which refers to all hardware, software, and procedures used to monitor and control industrial processes and devices [3]. Despite the use of these keys, the existing protocols are unable to satisfy to a large extent the key exchange and communication security properties (see, e.g., [4], [5]). A simple description of an OT environment is presented in Fig. 1. This figure shows the communication between devices or users in the OT environment. Note that the terms "device" and "user" are often used interchangeably in this paper.

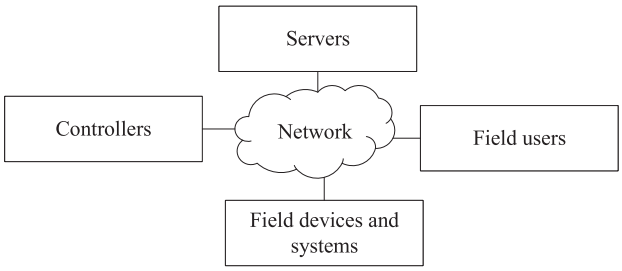

Fig. 1. A simple description of an OT environment.

Integration of the key exchange and communication security properties are still severely lacking in the OT. For example, WirelessHART protocol does not provide key secrecy as it lacks a dedicated key exchange protocol and data availability is a huge concern because session keys are issued to devices by a security manager. The need for a security manager makes it difficult to ensure key secrecy and further presents data availability concerns such as single point of failure [6]. Despite the expressiveness of the OT communication protocols, the lack of dedicated key exchange and security challenges of issuing and distributing session keys for secure communications have made a majority of the protocols unable to withstand cybersecurity issues such as unauthentication in the real world (see, e.g., [7]).

In this paper, we provide a universally composable key exchange framework for delivering the key exchange and communication security properties. It uses the Inexhaustible Interactive Turing Machine (IITM) model [8], [9] to design an ideal crypto-ops functionality F_{cl} for its cryptographic

operations on satisfying the key exchange and communication security properties. The IITM model is a universal composability model that facilitates modular reasoning and provides strong security notion for designing and analysing security protocols in the presence of an adversary. F_{cl} covers various cryptographic primitives, mainly an enhanced Elliptic Curve Diffie-Hellman key exchange (ECDH), i.e., Authenticated ECDH (AUT-ECDH), based on several factors such as identities to enhance the security of the key exchange. Note that ECDH is an Elliptic Curve Cryptography (ECC) [10] algorithm for key exchange. Our key contributions are as follows: (I) We propose the ideal functionality F_{cl} to support several cryptographic primitives used in our framework. F_{cl} can be realized under standard cryptographic assumptions, which are also applied to present and prove a realization P_{cl} of F_{cl}. (II) We analyse the WirelessHART protocol and find some security weaknesses in it. We use our framework to mitigate the weaknesses by presenting a new dedicated key exchange protocol, WirelessHARTKE, for the WirelessHART protocol.

II. Discussion and Related Work

Many communication protocols have been developed for communications between devices in the OT and each has tried to achieve specific security goals (see, e.g., [2], [11], [12], [13], [14], [15]). The Distributed Network Protocol 3 (DNP3) [11] for communications in process automation systems provides mutual authentication and supports the generation of cryptographic keys via its DNP3 Secure Authentication Version 6 (DNP3-SAv6) protocol layer. To approve and start the communications between two devices, the protocol relies on a one-time-use password. The distribution and reliance on the password make device enrolment and authorization as well as generation of session keys in the DNP3 vulnerable to security attacks that can compromise communications as the password can be inadvertently disclosed to an attacker.

WirelessHART protocol [2] is built without adequate key exchange properties being factored in (as briefly described in Section I), and another OT communication protocol, i.e., Process Field Bus (Profibus) [12], for monitoring measuring equipment, issues a token to devices for communication. However, Profibus does not support mutual authentication, key secrecy, and key confirmation between the devices. Furthermore, the security extension of an advanced version of Profibus, i.e., Process Field Net (Profinet) communication protocol [13], for handling larger messages and providing more speed and bandwidth describes data integrity and data confidentiality. However, Profinet lacks a dedicated key exchange to satisfy the key exchange properties.

The protocols proposed in [14] and [15] provide secure communication and data exchange in the energy grids. However, they do not offer key confirmation.

While many of the related OT communication protocols succeeded in providing some security, they suffer from one or more of the following drawbacks such as lack of dedicated key exchange and satisfying all the key exchange and communication security properties. In this paper, we propose a universally composable key exchange framework that satisfies all the OT key exchange and communication security properties and provides analysis in universally composable framework, which provides security in arbitrary adversarial environments (i.e., universal composition). We use our framework to enhance the WirelessHART protocol. Furthermore, we compare our enhancement with the existing related OT-based key exchange protocols [16], [17], [18] to show that it is suitable for the OT while providing universal composition. Note that: (I) Abbasinezhad-Mood and Nikooghadam [16] proposed a password-authenticated key exchange protocol for OT devices such as smart meters. However, the protocol rely on a pre-shared or one-time session key and the help of a service provider during key exchange thereby affecting key secrecy and data availability. (II) Srinivas et al. [17] and Garg et al. [18] proposed Elliptic Curve Cryptography (ECC) based key exchange protocols to enhance secure communications in smart grids. However, their protocols do not offer key confirmation and data confidentiality.

III. Universal Composability

In universal composability, real and ideal protocols are considered. An ideal protocol (or ideal functionality) stipulates the desired behaviour and intended security properties of a protocol, while the real protocol (or real functionality) i.e., the protocol to design and analyse, should realize the ideal protocol, i.e., it is supposed to be as secure as the ideal protocol. If there is a *real adversary* on the real protocol, then there should be an *ideal adversary* (or *simulator*) on the ideal protocol, such that any environment cannot differentiate the real and ideal settings. Note that universal composability has been utilised for designing and analysing cryptographic protocols in OT environments (see, e.g., [19], [20]).

In this paper, we use is the IITM model [21] with responsive environments from [9]. There are three different types of systems in the IITM model as follows: i) real and ideal protocols/functionalities; ii) adversaries and simulators; and iii) environments. Protocol systems and environment systems are systems that have an Input/Output (I/O) and network tapes or interfaces. Adversarial systems only have a network interface. We say that environment and adversarial systems are always responsive if they immediately respond to so-called *restricting messages* on the network. These messages can be represented in the form $(\text{Respond}, id, m)$, where id and m are random bit strings. In general, the use of restricting messages improves the expressivity of universal composability models.

Definition 1 [22]. *Let P and F be protocol systems with the same I/O interface, the real and the ideal protocol, respectively. Then, P realizes F ($P \leq F$) if and only if there exists an adversarial system S (an ideal adversary or a simulator) such that the systems P and $S|F$ have the same external interface and for all environment systems E, connecting only to the external interface of P (and hence, $S|F$), it holds true that $E|P \equiv E|S|F$, where the adversary in the real world is omitted by strong simulatability as it is subsumed by E.*

There are several composition theorems provided by the IITM model. One of the theorems (see below) handles the concurrent composition of a fixed number of protocol systems.

Theorem 1 [22]. *Let P_1, P_2, F_1, F_2 be protocol systems such that P_1 and P_2 as well as F_1 and F_2 only connect via their I/O interfaces with each other and $P_i \leq F_i$, for $i \in 1, 2$. Then, $P_1|P_2 \leq F_1|F_2$.*

IV. Proposed Framework

In this section, we present our framework, which comprises of the ideal crypto-ops functionality (F_{cl}) that provides cryptographic operations for modelling and satisfying the key exchange and communication security properties. F_{cl} allows its users to perform cryptographic operations in an idealized way. We say that if an OT communication protocol P uses F_{cl} for its cryptographic operations, then $P|F_{cl} \leq F$, where F is an ideal functionality, say an ideal key exchange functionality. Using the composition theorems, F_{cl} can be replaced with its realization after $P|F_{cl} \leq F$ has been proven. F_{cl} supports AUT-ECDH, which is similar to the ECDH, except that AUT-ECDH provides mutual authentication using several factors such as random numbers and identities.

A user of F_{cl} is identified by a tuple $(ID, lsid, r)$, where ID is a user identifier, $lsid$ is a local session identifier chosen and managed by cryptographic protocols (such as key exchange and communication protocols) in OT, and r is the role/tape that connects the user to F_{cl}, $lsid$, and ID. We assume that all users of F_{cl} are stored in a set Users. We parameterized F_{cl} with an ECDPGen(1^η) algorithm that is used to efficiently generate Elliptic Curve Domain (ECD) parameters. ECDPGen takes security parameter η as input and outputs ECD parameters (p, a, b, G, n, h), where p is a prime modulus, a and b are curve parameters, G is a generator point, n is an order of G in an elliptic curve EC over a finite field F_q, and h is a cofactor.

In F_{cl}, instead of returning secret keys to users, F_{cl} only returns pointers to the secret keys, which can be used by the users to perform many cryptographic operations. Nevertheless, every user gets a public share ps that belongs to a random secret rs, which are stored in a set RandSec. A random secret rs in RandSec is uncorrupted. Any random secret rs that cannot be identified by F_{cl} is stored in a set RandSec$_{un}$, which stores a list of all unidentified random secrets that cannot be used in F_{cl}. In this paper, "unidentified" random secrets can also be referred to as "corrupted" random secrets. Thus, the sets RandSec and RandSec$_{un}$ can be used to determine whether the keys created from the random secrets are uncorrupted and unidentified, respectively. Furthermore, F_{cl} also maintains the following sets: i) a set ERV of random values that may not be re-issued; ii) a set of ID&R of identities (e.g., ID) and random values (e.g., R) for linking the ID to R; iii) a set BlockedPS of public shares $ps = rs.G$ that may not be computed when rs is generated (i.e., $ps \notin$ BlockedPS); iv) a set ID&PS of identities (e.g., ID) and public shares (e.g., ps) for linking ID to ps; v) a set UnidentifiedKeys of unidentified keys that cannot be used for any cryptographic operations; vi) a set AUT-ECDHKeys of derived uncorrupted AUT-ECDH keys; vii) a set ConfirmationValues of confirmation values that guarantee that no other user has the same AUT-ECDH keys, except the two users that perform the key exchange; viii) a set MTexts of stored encrypted/MACed related information and symmetric keys; ix) a set PreSharedKeys of derived pre-shared keys; and x) a set SKeys of session keys. For brevity, all these sets are located in secure synchronised distributed databases in the OT. We assume that every user has real-time access to the databases for searching data.

We now initialise/activate F_{cl} for the first time and let it execute ECDPGen(1^η) and store the generated ECD parameters (p, a, b, G, n, h). The commands F_{cl} provides to a user $(ID, lsid, r)$ on the I/O interface are listed in Table I. Other commands such as MAC using pre-shared key $(\text{MAC}, ptr_{ii}, x)$, MAC Verification using pre-shared key $(\text{VMAC}, ptr_{ii}, x, y)$, and Session key derivation $(\text{DeriveSKey}, ptr_i, ptr_{ii})$ returns y, 1, and $(\text{SKeyPointer}, ptr_{iii})$, respectively to the user at the end of its execution, where $x = (R, R^{'}, ps, ps^{'})$, y is a valid MAC for x, and ptr_{iii} is a pointer pointing to a session key $k_{iii} = F_\eta^{'''}(hash(k_i, k_{ii}))$. The description of the three aforementioned commands is omitted due to page limit. Note that: (I) The ability to return messages directly to the user in F_{cl} without the help of any other user models *data availability*. (II) The execution of "MAC" or "VMAC" command models *data integrity*. (III) The execution of the "Session key derivation" command models *key secrecy*. (IV) The secret keys generated by F_{cl} can be applied to enhance mutual authentication.

We present a realization of F_{cl} denoted by P_{cl}. We construct P_{cl} by implementing all the operations of F_{cl} using standard cryptographic schemes. P_{cl} is a machine with the same network and I/O interfaces as F_{cl}. Upon the activation of P_{cl} for the first time by some message m, it initialises itself by executing ECDPGen and stores the results before processing m. Similar to F_{cl}, P_{cl} mantains random values, confirmation values, keys, and pointers to these keys in a distributed database within the functionality, and use them to determine the cryptographic primitive to execute.

V. Formal Security Verification

In this section, we simulate F_{cl} using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [23], which is widely used for automated security analysis of cryptographic protocols. We use the High-Level Protocol Specification Language (HLPSL) to implement F_{cl} and follow the Dolev-Yao attack model as the adversary model for describing the knowledge of the adversary. In the attack model, communication between users executing F_{cl} is performed over a public channel and an adversary can intercept, eavesdrop, modify, inject, replay, and delete data being transmitted between the users. Without loss of generality, pointers are not included and users do not share data with the adversary in our simulation. We use the On-the-fly Model Checker (OFMC) and Constraint Logic-based Attack Searcher (CL-AtSe) backends available in AVISPA to check for security attacks. As shown in Fig. 2, the simulation results from the backends confirm that our framework is safe and resilient against replay and man-in-the-middle attacks and the session key derived from it is safe from the attack model.

TABLE I
THE CRYPTOGRAPHIC COMMANDS OF THE IDEAL CRYPTO-OPS FUNCTIONALITY F_{cl}

Cryptographic Commands
Generate a new random value [(GenER)].The user $(ID, lsid, r)$ can request F_{cl} to generate a new random value. Upon receiving this request, F_{cl} checks whether $ID \in$ Users and returns (ER, $restricted$) if this check fails. Otherwise, it forwards the request to the adversary via a restricting message to provide a new random value $R \in \{1, ..., q-1\}$, where q is a large integer. F_{cl} ensures that R is new, i.e., $R \notin$ ERV. If this check fails, F_{cl} requests the adversary to provide another R until the check succeeds. Then, F_{cl} adds R to ERV and $(ID, R) \in$ ID&PS, and returns (ER, R) to the user.
Get generated ECD parameters [(GetECDP, R)]. The user $(ID, lsid, r)$ can request the parameter (p, a, b, G, n, h) that was generated during the initialisation of F_{cl}. Upon receiving this request, F_{cl} checks that $R \in$ ERV and and $(ID, R) \in$ ID&PS. If any of these checks fails, F_{cl} returns (ECDP, $restricted$) to the user. Otherwise, it sends (ECDP, (p, a, b, G, n, h)) to the user.
Generate a new random secret [(GenRandSec, R, (p, a, b, G, n, h))]. The user $(ID, lsid, r)$ can request F_{cl} to generate a pointer to a new random secret (rs). Upon receiving this request, F_{cl} checks that $R \in$ ERV and $(ID, R) \in$ ID&PS, and returns (RandSecPointer, $restricted$) to the user if any of the checks fails. Otherwise, F_{cl} forwards the request to the adversary via a restricting message to provide $rs \in \{1, ..., n-1\}$. Then, F_{cl} ensures that rs does not collide with existing random secrets (i.e., $rs \notin$ RandSec), and public share $ps = rs.G$ of rs has not been blocked from being generated (i.e., $ps \notin$ BlockedPS). If the check for rs fails, F_{cl} requests for another rs until the check succeeds. Then, F_{cl} adds rs to RandSec, ps to BlockedPS, and $(ID^{'}, ps^{'}) \in$ ID&PS, and then stores a pointer ptr pointing to rs for the user, and returns (RandSecPointer, ptr, ps) to the user.
Verify a public share [(VerifyPublicShare, $ID^{'}, ps^{'}$)]. The user $(ID, lsid, r)$ can request F_{cl} to verify a public share $ps^{'}$ of user $ID^{'}$. Upon receiving this request, F_{cl} checks that $ps^{'} \in$ BlockedPS and $(ID^{'}, ps^{'}) \in$ ID&PS, and returns (VerifyPublicShare, $restricted$) if any of these checks fail. Otherwise, it returns "$Okay$" (i.e., 1) to the user if the checks succeed. Note that the executions of VerifyPublicShare by users ID and $ID^{'}$ model *mutual authentication*.
Generate AUT-ECDH key [(GenAUT-ECDHKey, $R, ptr, ID^{'}, R^{'}, ps^{'}$)]. The user $(ID, lsid, r)$ can request F_{cl} to compute a new AUT-ECDH key k_i. Upon receiving this request, F_{cl} first checks that $R \in$ ERV, $(ID, R) \in$ ID&PS, $R^{'} \in$ ERV, and $(ID^{'}, R^{'}) \in$ ID&PS, and returns (AUT-ECDHKeyPointer, $restricted$) to the user if any of these checks fails. Otherwise, it creates a new pointer ptr_i as follows. If an AUT-ECDH key k_i has been previously generated using $ps^{'}$, $R^{'}$, and R, the pointer ptr_i is set pointing to k_i. Otherwise a new AUT-ECDH key k_i is generated as follows. • If $ps^{'}$ belongs to an unidentified random secret (i.e., $rs' \in$ RandSec$_{un}$), F_{cl} requests the adversary to provide an unidentified AUT-ECDH key k_i of type $t^{'} \in \{restricted\}$ via a restricting message (ProvideAUT-ECDHkey, $unidentified, rs^{'}, rs, R, R^{'}$) on the network. If an ECDH key k of type $t \in \{ecdhkey\}$ is provided, where $k = rs.ps^{'}$, and the AUT-ECDH key $k_i = F_\eta(hash(k, R, R^{'}))$ is computed using the hash function algorithm $hash(.)$ and Pseudo-Random Function (PRF) F provided by the adversary. Then, F_{cl} checks whether k_i is derived as $F_\eta(hash(k, R, R^{'}))$ and k_i is new (i.e., $k_i \notin$ AUT-ECDHKeys). If these checks succeed, it adds k_i to UnidentifiedKeys, sets the pointer ptr_i pointing to k_i for the user, and returns (AUT-ECDHKeyPointer, ptr_i) to the user. • If $ps^{'}$ belongs to an uncorrupted random secret (i.e., $rs' \in$ RandSec), F_{cl} requests the adversary to provide an uncorrupted AUT-ECDH key k_i of type $t_i \in \{aut-ecdhkey\}$ via a restricting message (ProvideAUT-ECDHkey, $uncorrupted, rs^{'}, rs, R, R^{'}$) on the network. If k_i provided by the adversary is not new (i.e., $k_i \in$ AUT-ECDHKeys) or k_i is not computed via $F_\eta(hash(k, R, R^{'}))$, F_{cl} requests for another k_i until a new k_i is provided. Then, F_{cl} adds k_i to AUT-ECDHKeys, sets the pointer ptr_i pointing to k_i for the user, and returns (AUT-ECDHKeyPointer, ptr_i) to the user.
AUT-ECDH Key Confirmation [(ConfirmAUT-ECDHKey, ptr_i)]. The user $(ID, lsid, r)$ can request F_{cl} to confirm an AUT-ECDH key k_i to which the pointer ptr_i points. Upon receiving this request, F_{cl} checks whether k_i is recorded for the user ID. If this check succeeds and no confirmation value cv has been recorded for k_i, it forward the request to the adversary via a restricting message to provide a new cv using a pseudo-random function $F^{'}$. F_{cl} ensures that cv is computed as $F^{'}_\eta(k_i)$. Then, it adds cv to ConfirmationValues and returns (AUT-ECDHKeyConfirmation, cv) to the user thereby satisfying *key confirmation*.
Pre-shared key derivation [(DerivePreSharedKey, ptr_i, cv)]. The user $(ID, lsid, r)$ can request F_{cl} to derive a new pre-shared key k_{ii} of type $t_{ii} \in \{preshared-key\}$. Upon receiving this request, F_{cl} checks that $cv \in$ ConfirmationValues and k_i is recorded for the user ID. If these checks succeed, it forwards the request to the adversary via a restricting message to provide such pre-shared key k_{ii}. Then, F_{cl} accepts k_{ii} under the following conditions as follows: i) if a pre-shared key k_{ii} has been recorded that it is derived using pseudo-random function $F^{''}$ as $F^{''}_\eta(hash(cv, k_i))$ such that $k_{ii} \in$ PreSharedKeys, F_{cl} stores a pointer ptr_{ii} pointing to k_{ii} for the user, and returns (PreSharedKeyPointer, ptr_{ii}) to the user; and ii) if no pre-shared key k_{ii} has been recorded that it is derived as $F^{''}_\eta(hash(cv, k_i))$, F_{cl} adds k_{ii} to $\{PreSharedKeys\}$, creates a pointer ptr_{ii} pointing to k_{ii} for the user, and returns (PreSharedKeyPointer, ptr_{ii}) to the user. Otherwise if $k_i \notin \{AUT-ECDHKeys\}$, F_{cl} asks the adversary via a restricting message to provide an unidentified pre-shared key k_{ii}, adds k_{ii} to UnidentifiedKeys, creates a pointer ptr_{ii} pointing to k_{ii} for the user, and returns (PreSharedKeyPointer, ptr_{ii}) to the user.
Encryption using pre-shared key [(Encryption, ptr_{ii}, u)]. The user $(ID, lsid, r)$ can request F_{cl} to encrypt a message $u = (R, R^{'}, ps, ps^{'})$ using the k_{ii} to which the pointer ptr_{ii} points. Upon receiving this message, F_{cl} checks whether k_{ii} to which ptr_{ii} points to is recorded for the user ID. If this check succeeds, it forwards the request to the adversary via a restricting message to encrypt u. Suppose v is the resulting ciphertext of u, F_{cl} verifies whether v can be decrypted using the decryption algorithm provided by the adversary. If this verification succeeds, it pairs (u, v), adds (u, v) to MTexts, and returns v to the user thereby satisfying *data confidentiality*. Otherwise, it returns (Encryption, $restricted$) to the user. Note that every decryption of an encryption in every reasonable encryption scheme will always produce the plaintext again.
Decryption using pre-shared key [(Decryption, ptr_{ii}, v)]. The user $(ID, lsid, r)$ can request F_{cl} to decrypt a ciphertext v using the key k_{ii} to which the pointer ptr_{ii} points to. Upon receiving this request, F_{cl} checks whether k_{ii} to which ptr_{ii} points to is recorded for the user ID. If this check succeeds and there exists exactly $(u, v) \in$ MTexts, F_{cl} returns u to the user. Otherwise, it returns (Decryption, $restricted$) to the user.

% OFMC % Version of 2006/02/13 SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SE SSIONS PROTOCOL /home/span/span/testsuite/results/ Framework.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s searchTime: 0.01s visitedNodes: 8 nodes depth: 3 plies	%CL-AtSe SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SE SSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/results/ Framework.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 2 states Reachable : 0 states Translation: 0.02 seconds Computation: 0.00 seconds

Fig. 2. Simulation Results using OFMC and CL-AtSe Backends.

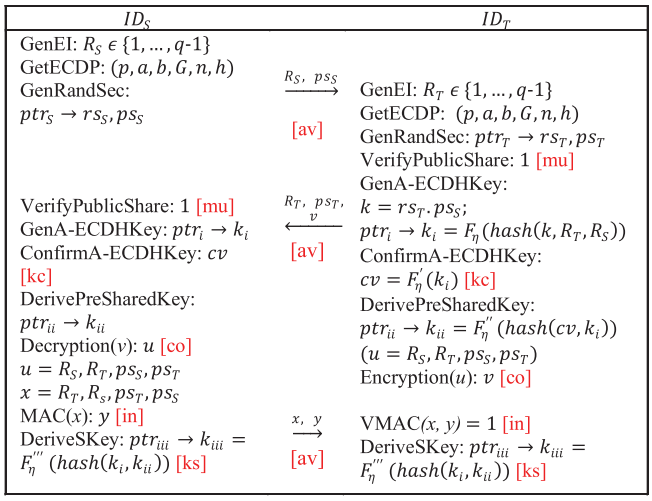

Fig. 3. WirelessHARTKE protocol. Abbreviations: mu — mutual authentication, kc — key confirmation, ks — key secrecy, av — data availability, co — data confidentiality, in — data integrity.

VI. Case Study

In this section, we demonstrate the usefulness of our framework. We analyse the WirelessHART protocol [2], which is a wireless mesh network communication protocol designed for industrial automation processes and applications. The essential components of the WirelessHART protocol include a network manager, a gateway, field devices, a security manager, and handheld devices. The security manager provides session keys, network keys, join key, and handheld keys to the network manager, which then distribute these keys to the devices - note that the WirelessHART protocol requires an end-to-end encryption and integrity check on encrypted data with the support of the distributed keys. We say that this form of key generation and distribution present serious security challenges on satisfying the key exchange and communication security properties considered in this paper. For example, data availability, data integrity (via the protocol's integrity check), and data confidentiality (via the protocol's end-to-end encryption) cannot be guaranteed since the security manager issues all keys. Furthermore, a device cannot guarantee key secrecy with another device. To see this, consider a setting where a security manager generates and outputs a secret key to initiator and responder instances. There is no (key secrecy) guarantee that the secret key is only known by the instances, and an attacker can easily use the key to send data packets by either compromising the security manager or the channel for distributing the key. As a consequence, all the key exchange properties would be violated. This is a direct attack against the WirelessHART protocol.

Furthermore, some security limitations of the WirelessHART protocol include lack of adequate and strong authentication and lack of key exchange. Using our framework, we address the above security challenges of the WirelessHART protocol. We model a dedicated key exchange protocol, WirelessHARTKE, for the WirelessHART protocol as depicted in Fig. 3. This figure shows that WirelessHARTKE protocol models *mutual authentication*, *key confirmation*, and *key secrecy* as well as *data availability*, *data integrity*, and *data confidentiality* using F_{cl}. If a secret key was not honestly generated by F_{cl}, it will be marked as corrupted. Thus, WirelessHARTKE protocol allows users to derive a session key in an idealized way. Additionally, users in the OT can now compute secret keys without the help of the security manager. In this case, unlike the WirelessHART, the WirelessHARTKE satisfies the key exchange properties and data availability. The following theorem states that the WirelessHARTKE protocol is a secure universally composable key exchange protocol for the WirelessHART protocol.

Theorem 2. *Let M_O and M_R be the machines modelling and satisfying the WirelessHARTKE protocol as described above, let F_{cl} and $F_{cl}^{'}$ be two versions of the ideal crypto-ops functionality with the same parameters. Then, the following holds true:*

$$M_O|M_R|F_{cl} \leq F_{cl}^{'}$$

Proof. We define a simulator S and show that $E|M_O|M_R|F_{cl} \equiv E|S|F_{cl}^{'}$ for all responsive environments $E \in Env(M_O|M_R|F_{cl})$. Let S internally simulates the protocol $M_O|M_R|F_{cl}$, where F_{cl} is used for modelling and satisfying the key exchange and communication security properties (as described in Section IV). As long as the environment is responsive, S is also responsive as it responds to restricting messages without any delay and it fulfills all runtime conditions.

We now argue that the simulation is perfect. Let $(ID_O, lsid_O, O)$ be an uncorrupted instance of the originator M_O. The Instance $(ID_O, lsid_O, O)$ wants to establish a key exchange session with user $ID^{'}$. S can simulate the behaviour of the instance $(ID_O, lsid_O, O)$ till it returns a session key. S finds the instance $(ID_R, lsid_R, R)$ of the responder R that can be paired with the instance $(ID_O, lsid_O, O)$. To return a session key pointer, S shows that the instance $(ID_O, lsid_O, O)$ had already accepted the second message of the WirelessHARTKE protocol and the session key of the user $(ID^{'})$ is uncorrupted. If the session key is corrupted, the protocol cannot continue. If some instance $(ID^{'}, lsid^{'}, r^{'})$ of

user $ID^{'}$ has encrypted a message $m = (R_S, R_T, ps_S, ps_T)$ and rs_T/ps_T is random secret/public share of $(ID^{'}, lsid^{'}, r^{'})$, then this instance is uncorrupted. As the user $ID^{'}$ remains uncorrupted, the instance $(ID^{'}, lsid^{'}, r^{'})$ cannot be corrupted by the adversary. Furthermore, as the instance $(ID^{'}, lsid^{'}, r^{'})$ considers the user ID_O as its key exchange partner that is recognised in the second message of the protocol, it does not consider itself corrupted. We now argue that the instance $(ID^{'}, lsid^{'}, r^{'})$ belongs to the responder R (or $r^{'}$ is a responder R): If $(ID^{'}, lsid^{'}, r^{'})$ is the instance of the originator O, then this implies that the second message of the protocol containing message $m = (R_T, R_S, ps_T, ps_S)$ was MACed by an uncorrupted instance of the originator ID_O. However, considering that rs_s/ps_s is ideally computed, the only instance that can MAC such message is $(ID_O, lsid_O, O)$ that does not return any MAC before accepting the second message of the protocol. This indicates that $r^{'}$ is a responder R and a MAC has been utilised to enhance integrity check on the encrypted message in the protocol.

By Theorem 1, F_{cl} can now be replaced with P_{cl} which yields that the WirelessHARTKE protocol is a secure universally composable key exchange protocol. Hence, compared with the present capabilities of key exchange in OT communication protocols [2], [11], [12], [13] and capabilities of the existing related key exchange protocols [16], [17], [18], we prove that WirelessHARTKE, using our framework, satisfies all the OT key exchange and communication security properties and proivides security in arbitrary adversarial environments.

VII. Conclusion and Future Work

In this paper, we have proposed a key exchange framework for enhancing the security of OT communication protocols. Our framework comprises an ideal crypto-ops functionality F_{cl} for cryptographic operations via universal composability. F_{cl} supports AUT-ECDH, which is based on ECDH and several factors such as random values and identities and is used to model key exchange and communication security properties. With the help of F_{cl}, OT communication protocols can be securely analysed and enhanced to mitigate cyber attacks. Our framework offers very high flexibility and modularity in the composition of cryptographic primitives. Using an OT case study, namely WirelessHART, we uncovered some weaknesses in it and then applied our framework to enhance its security by designing a dedicated key exchange protocol, namely WirelessHARTKE, for the WirelessHART. We show that the protocol offers strong universally composable security guarantees. In future work, we will apply our framework to enhance other OT communication protocols and provide its performance evaluation.

References

[1] A. S. Sani, D. Yuan, K. Meng, and Z. Y. Dong, "Kef: a key exchange framework for operational technology security standards and guidelines," in *2022 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2022, pp. 1–5.

[2] F. Group, "Hart communication protocol," 2022. [Online]. Available: https://fieldcommgroup.org/

[3] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6tisch: deterministic ip-enabled industrial internet (of things)," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 36–41, 2014.

[4] F. Luo, T. Feng, and L. Zheng, "Formal security evaluation and improvement of wireless hart protocol in industrial wireless network," *Security and Communication Networks*, vol. 2021, pp. 1–15, 2021.

[5] A. S. Sani, D. Yuan, and Z. Y. Dong, "Sdag: blockchain-enabled model for secure data awareness in smart grids," in *2023 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2023, pp. 1–5.

[6] S. Petersen and S. Carlsen, "Wirelesshart versus isa100.11a: The format war hits the factory floor," *IEEE Industrial Electronics Magazine*, vol. 5, no. 4, pp. 23–34, 2011.

[7] "ICSA-20-205-01: Schneider Electric Triconex Tristation and Tricon Communication Module," 2020. [Online]. Available: https://us-cert.cisa.gov/ics/advisories/icsa-20-205-01

[8] R. Küsters, M. Tuengerthal, and D. Rausch, "The iitm model: a simple and expressive model for universal composability," *Journal of Cryptology*, vol. 33, pp. 1461–1584, 2020.

[9] J. Camenisch, R. R. Enderlein, S. Krenn, R. Küsters, and D. Rausch, "Universal composition with responsive environments," in *Proceedings, Part II, of the 22Nd International Conference on Advances in Cryptology — ASIACRYPT 2016 - Volume 10032*. New York, NY, USA: Springer-Verlag New York, Inc., 2016, pp. 807–840.

[10] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.

[11] D. U. Group, "Overview of DNP3 protocol," 2023. [Online]. Available: https://www.dnp.org/About/Overview-of-DNP3-Protocol

[12] PI, "Overview - Profibus," 2023. [Online]. Available: https://www.profibus.com/technology/profibus/overview/

[13] PI, "Overview - Profinet," 2023. [Online]. Available: https://www.profibus.com/technology/profinet/overview/

[14] A. S. Sani, D. Yuan, P. L. Yeoh, J. Qiu, W. Bao, B. Vucetic, and Z. Y. Dong, "Cyra: A real-time risk-based security assessment framework for cyber attacks prevention in industrial control systems," in *2019 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2019, pp. 1–5.

[15] A. S. Sani, D. Yuan, S. Ogaji, and Z. Y. Dong, "Cyreume: A real-time situational awareness and decision-making blockchain-based architecture for the energy internet," in *Handbook of Real-Time Computing*. Springer, 2022, pp. 787–835.

[16] D. Abbasinezhad-Mood and M. Nikooghadam, "Efficient anonymous password-authenticated key exchange protocol to read isolated smart meters by utilization of extended chebyshev chaotic maps," *IEEE Transactions on Industrial Informatics*, 2018.

[17] M. Jo, S. Jangirala, A. K. Das, X. Li, and M. K. Khan, "Designing anonymous signature-based authenticated key exchange scheme for iot-enabled smart grid systems," *IEEE Transactions on Industrial Informatics*, 2020.

[18] S. Garg, K. Kaur, G. Kaddoum, J. J. Rodrigues, and M. Guizani, "Secure and lightweight authentication scheme for smart metering infrastructure in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3548–3557, 2019.

[19] A. S. Sani, D. Yuan, W. Bao, Z. Y. Dong, B. Vucetic, and E. Bertino, "Universally composable key bootstrapping and secure communication protocols for the energy internet," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2113–2127, 2019.

[20] A. S. Sani, D. Yuan, K. Meng, and Z. Y. Dong, "R-chain: a universally composable relay resilience framework for smart grids," in *2021 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2021, pp. 01–05.

[21] R. Kusters, "Simulation-based security with inexhaustible interactive turing machines," in *Proceedings of the 19th IEEE Workshop on Computer Security Foundations*, ser. CSFW '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 309–320.

[22] R. Küsters, M. Tuengerthal, and D. Rausch, "Joint state composition theorems for public-key encryption and digital signature functionalities with local computation," *Journal of Cryptology*, vol. 33, pp. 1585–1658, 2020.

[23] "Avispa," 2003. [Online]. Available: http://www.avispa-project.org/