KEF: A Key Exchange Framework for Operational Technology Security Standards and Guidelines

Abubakar Sadiq Sani*, Dong Yuan†, Ke Meng‡, and Zhao Yang Dong‡
* School of Computing and Mathematical Sciences, University of Greenwich, London, United Kingdom
Email: s.sani@greenwich.ac.uk
† School of Electrical and Information Engineering, The University of Sydney, Sydney, Australia
Email: dong.yuan@sydney.edu.au
‡ School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia
Email: [ke.meng, joe.dong]@unsw.edu.au

***Abstract*—Recent findings show that many security standards and guidelines for Operational Technology (OT) in smart grids do not satisfy all key exchange properties such as mutual authentication, key secrecy, and key confirmation. As accepted best practices are undergoing tailoring due to increase in remote grid operations that have also led to an increase in cyber attacks against smart grids, we propose to enhance key exchange in the OT security standards and guidelines via KEF, a key exchange framework for satisfying and enforcing the key exchange security properties to mitigate cyber attacks. KEF comprises a set of cryptographic operations and a set of key exchange states for key exchange operations. We analyse the security of KEF using Automated Validation of Internet of Security Protocols and Applications (AVISPA) tool and demonstrate its security benefits by applying it to a real-world key establishment scheme, Special Publication (SP) 800-56A Revision 3, of the National Institute of Standards and Technology (NIST).**

***Index Terms*—Operational Technology, security, key exchange, standards, guidelines.**

I. Introduction

Operational Technology (OT) refers to the physical grid assets and applications used to monitor, control, and manage smart grid operations [1]. To secure data and communications in the OT, key exchange is one of the most widely recommended data and communications security methods in the OT security standards and guidelines such as the International Electrotechnical Commission (IEC) 62351 [2] and National Institute of Standards and Technology (NIST) Special Publication (SP) 800-82 [3], respectively. The essential properties for key exchange are mutual authentication, key confirmation, and key secrecy which guarantee that a shared secret session key is shared between intended devices, the devices have the same shared key, and no other device knows about the key, respectively. Despite the use of key exchange, many of the existing OT security standards and guidelines have at least one of the following limitations: (A) Lack of satisfying and enforcing all the key exchange security properties [2], [3], [4]; and (B) Relying on authenticators such as Public Key Infrastructure (PKI) certificates and passwords issued by a trusted entity like a certificate authority (CA) [5]. These limitations are complex security concerns and increasingly challenging in the ongoing efforts of tailoring accepted best practices to provide remotely secure proactive insight into grid operations.

The OT security standards and guidelines can be viewed as an amalgamation of multiple security methods and sub-methods such as key exchange and key confirmation, respectively. However, satisfying all these methods are still severely lacking in the OT security standards and guidelines. For example, NIST SP 800-57 guideline [4] does not provide key secrecy when a pre-shared key is used to derive a shared secret session key because the pre-shared key is issued by a trusted authority thereby making it difficult to ensure key secrecy.

Despite the expressiveness of the OT security standards and guidelines, the lack of enforcing all the key exchange security properties have made a majority of the standards and guidelines unable to withstand the fast-growing cyber attacks against the grid infrastructure. Therefore, one of the main contributions in this paper is to provide a key exchange framework that enhances the OT security standards and guidelines with all the key exchange security properties to mitigate cyber attacks.

In this work, we provide a key exchange framework called KEF, for delivering the key exchange security properties in a secure manner. It provides several cryptographic operations to satisfy the key exchange security properties. Additionally, KEF covers various cryptographic primitives, mainly an Attribute-based Elliptic Curve Diffie-Hellman key exchange (A-ECDH) based on the Elliptic Curve Cryptography (ECC) algorithm [6] (i.e., ECDH) and devices attributes such as an ephemeral random value, which is generated to strengthen the security of the key exchange. More specifically, our contributions are as follows: i) We propose a set of cryptographic operations $KEF.C$ used in KEF; ii) We propose KEF, which contains a set of key exchange states $KEF.S$ that utilises $KEF.C$ and enforces the key exchange security properties; iii) We analyse the security of KEF using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [7]; and iv) We illustrate the usefulness of KEF by analysing and enhancing the Category of two ephemeral key pairs (C(2e,0s)) key establishment scheme of the NIST SP 800-56A Revision 3 [8], which we refer to as "NIST.SP.800-56A.Rev3.C(2e,0s)" in this work.

II. Related Work

Many of the existing OT security standards and guidelines (see, e.g., [2], [3], [4], [5], [9], [10]) have different advantages (such as providing authentication) and shortcomings (such as lack of key secrecy). The IEC 62351-9 standard [2] describes key management methods such as authentication and authorisation for protecting messages in power systems. While the standard supports key exchange, it relies on a key server for issuing and managing cryptographic keys thereby impacting key secrecy. The IEEE 1686 standard [5] describes cybersecurity capabilities for Intelligent Electronic Devices (IEDs), which are widely deployed in the OT to enable advanced power automation. The standard supports key exchange to achieve secure communications between IEDs. While the IEDs can authenticate any configuration software used to access or change their configurations, the software is not equipped with authenticating the IEDs thereby leading to the absence of mutual authentication. Additionally, key confirmation and key secrecy are not captured as techniques employed by the standard. The IEC TS 60870-5-7 standard [9] describes secure authentication in power system monitoring, control, and communications. While the standard supports secure authentication, utilising the standard with its symmetric keys does not satisfy mutual authentication because only one predefined device is authenticated during monitoring, control, and communications. In this case, the standard cannot guarantee that the symmetric keys are utilised by intended devices.

The NIST SP 800-82 guideline [3] on securing Industrial Control Systems (ICS) supports control systems security. It also supports authentication via a secret code known to the devices in advance. This form of authentication shows that the NIST SP 800-82 guideline lacks key secrecy as the secret code can be exploited by an adversary to compromise the devices, which supports (OT-based) grid operations such as power supply. The NIST SP 800-57 guideline [4] describes the recommendation for key management and supports key confirmation and identity authentication. It also supports several security features such as cryptographic algorithms (like cryptographic hash functions and symmetric-key algorithms) and policy security planning. As the guideline recommends secret keys distribution, such a distribution process impacts key secrecy. The NIST SP 800-56B guideline [10] describes the recommendation for pair-wise key establishment. It supports key confirmation and source authentication for key pair. As the guideline relies on a Rivest–Shamir–Adleman (RSA) key with pre-shared secret values to establish shared secret keys and secret keying materials, this reliance affects key secrecy. While many of the above OT security standards and guidelines succeeded in providing some security, they do not satisfy and enforce all the key exchange security properties.

III. Preliminaries

A. OT Cybersecurity Architecture

OT cybersecurity architecture supports grid assets, processes, and other devices (such as sensors) with the security required for secure grid operations. Through firewall, Intrusion Prevention System (IPS), and Intrusion Detection System (IDS) as illustrated in Fig. 1, the architecture can be designed and implemented to provide secure remote grid operations. Every sensor measures a physical property of some grid asset and process and sends the measurement to a controller, which interprets the received information and transmits the interpretation to an actuator for manipulation. All the information or data are available at the Human Machine Interface (HMI), which allows the remote operators to interact with the grid assets and other devices. The grid asset, sensor, controller, actuator, and HMI are vulnerable as the data transmitted and wireless communication channels between the source (say, the grid asset) and the destination (say, the sensor) can be manipulated. Thus, it is necessary to perform a key exchange with mutual authentication, key secrecy, and key confirmation between all the entities before any data exchange in the OT. To maintain business continuity, owners of the grid have been assembling software that can securely run remotely across firewalls to support grid operations and enable seamless collaborations among grid personnel such as operations and engineering. The firewalls provide an interface between the remote operators and grid assets that are utilised in the OT. The interface can be prevented from fast-growing cyber attacks by using KEF (see below) to establish a key exchange that satisfies and enforces the key exchange security properties.

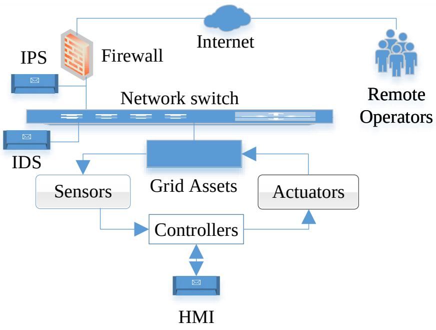


Fig. 1. A Simple OT Cybersecurity Architecture.

B. Attack Model

We follow the Dolev-Yao threat model [11] as our attack model for describing the knowledge of an adversary that can eavesdrop, intercept, and manipulate the data transmitted and wireless communication channels between the entities. The Dolev-Yao threat model is suitable for analysing our key exchange security properties and it has been widely accepted as a standard threat model for key exchange in smart grids.

C. Cybersecurity Goals

1) Mutual Authentication: It is paramount that devices authenticate each other since data collected in the OT are utilised for many purposes such as monitoring grid operations.

2) Key Secrecy: The use of pre-shared keys (issued by a server) and PKI certificates (issued by a CA) can immediately destroy the secrecy of the pre-shared keys since the server and authorities can be compromised by an adversary and present a single point of failure. The secrecy of secret keys in the OT can be prevented from any compromise using KEF (see Sections IV and V for more details).

3) Key Confirmation: Assurance that the users can compute and hold the same secret key supports the security of data exchanged in the OT. In this case, we introduce explicit and implicit key confirmation in this work to provide assurances on the shared secret key and its computation, respectively.

Other cybersecurity goals include confidentiality, integrity, availability, and perfect forward secrecy.

IV. The KEF Framework

In this section, we present KEF, which offers a set of cryptographic operations (denoted as $KEF.C$) and a set of key exchange states (denoted as $KEF.S$) for satisfying and enforcing mutual authentication, key secrecy, and key confirmation. $KEF.C$ supports an A-ECDH, which is similar to the ECDH, except that A-ECDH provides mutual authentication and key confirmation using users attributes. Note that in what follow we use the term "user" to refer to an OT device like a sensor and therefore the terms "user" and "device" are often used interchangeably in this work.

A. Cryptographic Operations in KEF (KEF.C)

A user of $KEF.C$ is identified by an identity ID that is stored in a subset Users (of $KEF.C$). We parameterized $KEF.C$ with elliptic curve domain 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. $KEF.C$ guarantees that only the genuine owners of A-ECDH keys can use the keys to derive shared secret session keys for secure data exchange in the OT. In $KEF.C$, we implement all the cryptographic operations using common standard cryptographic schemes in a natural and expected way. Furthermore, $KEF.C$ also utilises another subset BlockedPS of identities (e.g., ID) and public shares or public keys (e.g., $ps = rs.G$), which may not be computed when a random secret (or private key) rs is selected (i.e., $(ID, ps) \in$ BlockedPS), to prevent random secret collisions. For brevity, we say that $KEF.C$ and its subsets are located in secure synchronised distributed databases of OT authorities that are responsible for registering users in the OT. We assume that every user in the subset Users has real-time access to the databases, which share identical information. The list of cryptographic operations (Cs) offered to a user ID (or another user $ID^{'}$) by $KEF.C$ are as follows.

- **Get elliptic curve domain parameters (C-1)**. The user ID can get the (p, a, b, G, n, h). In this case, the user gets the (p, a, b, G, n, h) that is available in $KEF.C$.
- **Select a new ephemeral random value (C-2)**. The user ID can select a new ephemeral random value $R \in \{1, ..., q-1\}$, where q is a large integer.
- **Select a new random secret (C-3)**. The user ID can select a new random secret $rs \in \{1, ..., n-1\}$ and compute a public share ps using the domain parameters (p, a, b, G, n, h) by computing $ps = rs.G$. Then, the user adds (ID, ps) to the set BlockedPS.
- **Verify a public share (C-4)**. The user ID can verify a received public share $ps^{'}$ of user $ID^{'}$ by checking that $(ID^{'}, ps^{'}) \in$ BlockedPS. If the check succeeds, the user selects a verification value $vv \in \{1, ..., w-1\}$ to support mutual authentication, where w is a large integer.
- **Generate A-ECDH key (C-5)**. The user ID can generate an A-ECDH key k_i from its random secret rs and ephemeral random value R and some public share $ps^{'}$ and ephemeral random value $R^{'}$ of a user $ID^{'}$. If $(ID^{'}, ps^{'}) \in$ BlockedPS, the user first compute an ECDH key $k = rs.ps^{'}$ and then compute an A-ECDH key $k_i = hash(k.R.R^{'})$ to support A-ECDH key secrecy, where $hash(.)$ is a cryptographic hash function algorithm.
- **Implicit A-ECDH key confirmation (C-6)**. The user ID can be assured that the user $ID^{'}$ can compute an A-ECDH key k_i by computing some messages $M_1 = Enc_{k_i}(vv, ps, ps^{'}, R^{'}, R)$ and $M_2 = MAC_{k_i}(M_1)$, a verification key $k_{ii} = Hash(k_i, vv)$, and additional messages $M_3 = Enc_{k_{ii}}(R, R^{'})$ and $M_4 = MAC_{k_{ii}}(M_3)$, where $Enc(.)$ is the encryption part of an Advanced Encryption Standard (AES) algorithm and $MAC(.)$ is a MAC algorithm.
- **Explicit A-ECDH key confirmation (C-7)**. The user $ID^{'}$ can be assured that the user ID holds an A-ECDH key k_i by computing $VMAC_{k_i}(M_2) = M_1$, $Dec_{k_i}(M_1) = vv, ps, ps^{'}, R^{'}, R$, $k_{ii} = Hash(k_i, vv)$, $VMAC_{k_{ii}}(M_4) = M_3$, and $Dec_{k_{ii}}(M_3) = R, R^{'}$, where $VMAC(.)$ is a MAC verification algorithm and $Dec(.)$ is the decryption part of the AES algorithm. The successful computations of M_3 and M_4 confirm that ID holds the A-ECDH key k_i.
- **Compute shared secret session key (C-8)**. The user ID can compute a shared secret session key k_{iii} from an A-ECDH key k_i, a verification key k_{ii}, and verification values vv and $vv^{'}$ by computing $k_{iii} = F_\eta(hash(k_i, k_{ii}, vv, vv^{'}))$ to support shared secret session key secrecy, where F_η is a secure Pseudo-Random Function (PRF).

Note that all the outputs associated with mutual authentication, key secrecy, and key confirmation operations are utilised in the computation of the shared secret session key to satisfy all the key exchange security properties.

B. Key Exchange States in KEF (KEF.S)

We present $KEF.S$, which enforces all the key exchange security properties, enhances identity verification, and provides assurance of users attributes during key exchange. $KEF.S$ uses an interactive key exchange program KE that can be executed by users. KE uses $KEF.C$ as a subroutine to perform cryptographic operations. To determine the key outputted after a successful key exchange, we parameterized KE with encryption schemes, a MAC scheme, three families of PRFs (that take a key and a salt as input and output another key), and the (p, a, b, G, n, h) that is similar to $KEF.C$.

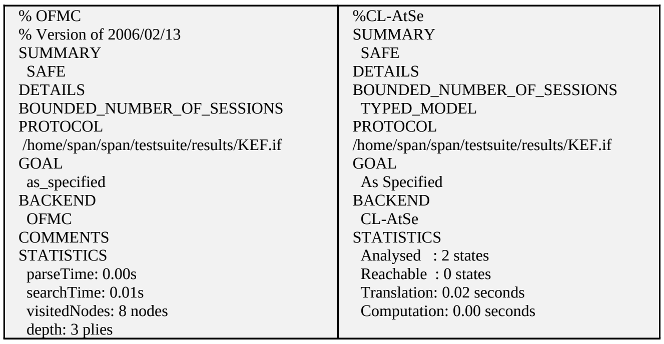
A user of KE is identified as (ID, sid), where ID is a user identity and sid is a session identifier. In this work, we assume that session identifiers are public and determined by incoming and outgoing messages of the associated sessions to simplify the use of KE. All messages from/to KE are prefixed with (ID, sid). KE maintains key exchange states restricted-KE, begin-KE, in-KE, finishing-KE, and finished-KE, and ensures that a user is in a session with its intended partner. The state of every user is initially set to restricted-KE. The operations provided by KE are as follows.

- A user (ID, sid) with $state(ID, sid)$ = restricted-KE can use KE to start a key exchange by sending a key exchange request $m =$ (InitialiseKE, $ID^{'}$), where $ID^{'}$ is user identifier of intended responder. Upon receiving this request, KE uses $KEF.C$ to verify the ID and $ID^{'}$ of the users (i.e., $ID \in$ Users and $ID^{'} \in$ Users) and that both users are not yet in a key exchange session. If the verifications succeed, KE sets $state(ID, sid) :=$ begin-KE, sets $partner(ID, sid) := (ID^{'}$, begin-KE), stores that (ID, sid) and $(ID^{'}, sid)$ are in the same key exchange session, and returns $Okay$ to the user, thus providing a guarantee of creating a unique session of authenticated key exchange partners. Note that as there is no any trusted authority or third party between ID and $ID^{'}$, KE models *availability*.
- A user (ID, sid) with $state(ID, sid)$ = begin-KE can use KE to access the following cryptographic operations of $KEF.C$: i) C-1; ii) C-2; iii) C-3; and iv) C-4. Once KE sends a verification value vv to the user, where vv represents the execution result of the C-4 operation, it sets the state of the user as $state(ID, sid) :=$ in-KE. Thus, C-4 is the last operation that the user can execute in the state begin-KE, where KE models *mutual authentication* via verification of users attributes. Note that: i) the users ID and $ID^{'}$ send (R, ps) and $(R^{'}, ps^{'})$, respectively, to each other to support KE in this state; and ii) mutual authentication is achieved via verification of (R, ps) and $(R^{'}, ps^{'})$ by users $ID^{'}$ and ID, respectively.
- A user (ID, sid) with $state(ID, sid)$ = in-KE can use KE to access the following cryptographic operations of $KEF.C$: i) C-5; ii) C-6; and iii) C-7. Once KE forwards the execution result of the C-7 operation to the user, it sets the state of the user as $state(ID, sid) :=$ finishing-KE. KE models *integrity*, *confidentiality*, and *key confirmation* after the execution of the operations C-5, C-6, and C-7, respectively. Note that the users ID and $ID^{'}$ send (M_1, M_2, M_3, M_4) and $(M_1^{'}, M_2^{'}, M_3^{'}, M_4^{'})$, respectively, to each other to support KE in this state, where $M_1 = M_1^{'}$, $M_2 = M_2^{'}$, $M_3 = M_3^{'}$, and $M_4 = M_4^{'}$ (see Section IV-A for detailed description of M_1, M_2, M_3, and M_4). In $M_1^{'}$, $M_2{'}$, $M_3^{'}$, and $M_4^{'}$, $ID^{'}$ replaces the inputs in the messages with its own corresponding values, for example ps is replaced with $ps^{'}$.
- A user (ID, sid) with $state(ID, sid)$ = finishing-KE can use KE to access the C-8 operation of $KEF.C$. KE sets the state of the user as $state(ID, sid) :=$ finished-KE as soon as it sends the execution result of the C-8 operation to the user thereby guaranteeing the creation of a unique shared secret session key. Thus, KE models *key secrecy* after executing the C-8 operation.

Note that: (I) KE should be widely applicable to key exchange in the OT security standards and guidelines by using the messages exchanged between users ID and $ID^{'}$ and $KEF.C$ operations. These messages and operations can be applied iteratively to enhance the key exchange. (II) As session keys from real-world key exchange protocols expire within a short time, all session keys derived using KEF are also short-lived. (III) Since ID and $ID^{'}$ do not have access to any keys or cryptographic operations after finalising the key exchange (or in the state finished-KE), we say that KEF models *perfect forward secrecy*, which give assurances that the A-ECDH key, shared secret session key, and cryptographic operations will not be compromised after key exchange.

V. Formal Security Verification

In this section, we simulate and verify the security of our framework using the widely-accepted security tool, AVISPA [7], which is used for automated security analysis of cryptographic schemes in smart grids. We implement KEF using the High-Level Protocol Specification Language (HLPSL) to model the roles of the key exchange users, set of key exchange states, set of cryptographic operations, attacker knowledge in KEF, and secrecy and authentication goals in KEF. Furthermore, we follow our attack model as described in Section III in the modellings. In our simulation, we used the On-the-fly Model Checker (OFMC) and Constraint Logic-based Attack Searcher (CL-AtSe) backends available in AVISPA to check for security attacks against KEF. More details about our simulation are as follows: (I) We specify $KEF.C$ and $KEF.S$ which are decomposed into two roles, users ID_S and ID_T. (II) We describe the communications (i.e., number of steps) and exchanged messages between ID_S and ID_T using KEF, i.e., $KEF.C$ and $KEF.S$, as follows: i) Step 1 $-$ $ID_S \rightarrow ID_T = (R_S, ps_S)$; ii) Step 2 $-$ $ID_T \rightarrow ID_S = (R_T, ps_T)$; iii) Step 3 $-$ $ID_S \rightarrow ID_T = (M_1, M_2, M_3, M_4)$; and iv) Step 4 $-$ $ID_T \rightarrow ID_S = (M_1^{'}, M_2^{'}, M_3^{'}, M_4^{'})$ (see Section IV for detailed description of the messages). (III) We describe the sessions of KEF and model the attacker knowledge according to KEF as presented in Section III. As shown in Fig. 2, the simulation results from the OFMC and CL-AtSe backends confirm that KEF is safe and resilient against replay and man-in-the-middle attacks and securely satisfies the key exchange security properties. Hence, the derived shared secret session key is safe from the Dolev-Yao attack model. II. It should be noted that though our framework is proved to be resilient against the Dolev-Yao attack model, other unknown attack models which cannot be anticipated may disrupt the execution of the proposed framework, thus, we assume that the proposed framework cannot be utilised if any unknown attack model disrupts its execution.



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/KEF.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_SESSIONS
  TYPED_MODEL
PROTOCOL
/home/span/span/testsuite/results/KEF.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed   : 2 states
  Reachable  : 0 states
  Translation: 0.02 seconds
  Computation: 0.00 seconds
```

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

VI. Case Study

We carry out a case study to demonstrate the usefulness and application of KEF by analysing and enhancing the NIST.SP.800-56A.Rev3.C(2e, 0s) scheme [8], which is illustrated in Fig. 3. The scheme is based on ECDH and meant to provide mutual authentication and key secrecy but does not specify key confirmation incorporation. Each user of the scheme generates an ephemeral key pair, i.e., rs and ps, and no static key pairs are used. We point out a weakness in the scheme and incorporate key confirmation for the scheme. We argue that the scheme does not offer mutual authentication. To show this, we consider a setting where ID_S computes a shared secret key k_{iii} that was generated from its ps_s and ID_T's ps_T. ID_T that sent ps_T might have received a different public share, say $ps_s^{'} \neq ps_s$, in the first message of the scheme. If $ps_s^{'}$ is not generated via $KEF.C$, then it cannot be verified, thus, the key k_{iii} from ps_s/ps_T and all keys derived from it cannot be computed via $KEF.C$. To fix this problem, an ephemeral random value R_S is required in the first message of the scheme and an ephemeral random value R_T, public share ps_T, and MAC MAC_{k_i} of (R_T, R_S, ps_s, ps_T) are required in the second message of the scheme to support mutual authentication, where k_i is an A-ECDH key.

Furthermore, we argue that the scheme is vulnerable to *unknown key-share attack*. To support our argument, we consider a setting where an adversary I intercepts the first message of the scheme from ID_S, replaces ps_s with ps_I, and then forwards the message to ID_T. At the end of the scheme, ID_T mistakenly believes the k_{iii} is shared with $I \neq ID_S$. To fix this, we use the notion similar to the above via our framework, i.e., R_S and $(R_T, Enc_{k_i}(values), MAC_{k_i}(Enc_{k_i}(values))$ are required in the first and second messages of the scheme, respectively, where $values$ is (R_T, R_S, ps_s, ps_T). Interestingly, these fixes or additions (via implementing C-4, C-5, C-6, and C-7 operations of $KEF.C$ as described in Section IV) enhance mutual authentication and incorporate key confirmation for the scheme, which we now referred to an Enhanced NIST.SP.800-56A.Rev3.C(2e, 0s) scheme as a result of the additions. Note that KEF is modular and thus the messages exchanged and cryptographic operations in it can be separated, combined, and applied iteratively to construct more secure key exchange schemes in the OT security standards and guidelines.

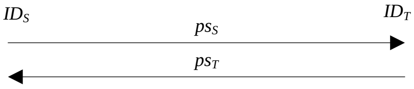


Fig. 3. The NIST.SP.800-56A.Rev3.C(2e, 0s) scheme.

VII. Conclusion and Future Work

In this paper, we have proposed an OT key exchange framework KEF that satisfies and enforces key exchange security properties such as mutual authentication, key secrecy, and key confirmation in smart grids. KEF comprises a set of cryptographic operations $KEF.C$, which supports the A-ECDH that enhances the existing ECDH with mutual authentication and key confirmation. It also comprises of a set of key exchange states $KEF.S$, which uses $KEF.C$ for cryptographic operations and a key exchange program KE for enforcing the key exchange security properties. As accepted best practice security are undergoing tailoring to provide highly secure remote operations, key exchange in many of the OT security standards and guidelines can be enhanced using KEF. The security analysis of KEF shows that it is a secure key exchange framework. We have demonstrated the usefulness of KEF in a case study, where we uncovered some weaknesses and provided an enhancement using KEF. In future work, we will extend KEF to include key exchange privacy, introduce its comparison with other key exchange frameworks, provide its performance analyses and experiments, and discuss its deployment process in the OT.

References

[1] D. Hough, “Iet: cyber security in modern power systems: It and operational technology integration,” in *IET Cyber Security in Modern Power Systems*. IET, 2016, pp. 1–21.

[2] “Power systems management and associated information exchange - Data and communications security - Part 9: Cyber security key management for power system equipment,” International Electrotechnical Commission, Standard IEC 62351-9, 2017.

[3] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, “Guide to industrial control systems (ics) security (nist special publication 800-82 rev. 2),” *Gaithersburg, MD: US National Institute of Standards and Technology*, vol. 10, 2015.

[4] E. Barker, “NIST Special Publication 800-57 part 1, Revision 5 (Recommendation for Key Management: Part 1 - General),” *NIST, Tech. Rep*, 2020.

[5] “IEEE Standard for Intelligent Electronic Devices Cyber Security Capabilities,” IEEE, Standard IEEE Std 1686-2013, 2014.

[6] A. Koblitza, N. Koblitzb, and A. Menezes, “Elliptic curve cryptography: The serpentine corse of a paradigm shift,” *Journal of Number Theory*, vol. 131, no. 5, p. 781–814, 2011.

[7] “Automated validation of internet security protocols and applications,” 2003. [Online]. Available: http://www.avispa-project.org/

[8] E. Barker, L. Chen, A. Roginsky, A. Vassilev, and R. Davis, *Recommendation for pair-wise key establishment schemes using discrete logarithm cryptography*. National Institute of Standards and Technology, 2018.

[9] “Telecontrol equipment and systems - part 5-7: Transmission protocols - security extensions to IEC 60870-5-101 and IEC 60870-5-104 protocols,” International Electrotechnical Commission, Standard IEC TS 60870-5-7, 2013.

[10] E. Barker, L. Chen, A. Roginsky, and A. Vassilev, “Nist special publication 800-56b. recommendation for pair-wise key establishment using integer factorization cryptography,” 2019.

[11] D. Dolev and A. Yao, “On the security of public key protocols,” *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.