A Universally Composable Key Exchange

Protocol for Advanced Metering Infrastructure

in the Energy Internet

Abubakar Sadiq Sani [*,*](https://orcid.org/0000-0001-8201-8770) *Member, IEEE*, Dong Yuan*, Member, IEEE*, Wei Bao [*,*](https://orcid.org/0000-0003-1874-1766) *Member, IEEE*, and Zhao Yang Dong [*,*](https://orcid.org/0000-0001-9659-0858) *Fellow, IEEE*

***Abstract*—The increasing adoption of multiway communications in the advanced metering infrastructure (AMI) of the energy Internet, which is known as the Internet-based smart grid, raises a new question about the security of customers’ sensitive data and how the data can be protected from growing cyber attacks such as side-channel and false data injection attacks. The dynamic nature of remote connect/disconnect of components in the AMI also brings new types of security threats. To achieve secure multiway communications and remote connect/disconnect of components, the AMI requires a key exchange protocol (KEP) that meets a number of its security requirements such as confidentiality, integrity, availability, identification, authentication, and access control. In this context, in this article we present a KEP that uses an ideal crypto functionality and an ideal AMI key exchange functionality based on universal composability, which allows modular design and analysis of cryptographic protocols. The former functionality enables AMI components or users to perform authenticated cryptographic operations, while the later functionality enables the users to meet the AMI security requirements before generating a shared secret session key, which can be used in an ideal manner. We carry out experiments to validate the performance of our protocol, and the results show that our protocol offers better performance benefits compared to the existing related protocols and is suitable for the Energy Internet. We further demonstrate the usefulness of our ideal functionalities as a security reinforcement for a widely used KEP, namely the Elliptic Curve Diffie–Hellman.**

***Index Terms*—Advanced metering infrastructure (AMI), Energy Internet, key exchange, security, universal composability.**

|  |
| --- |
| Manuscript received March 21, 2019; revised November 29, 2019; accepted January 14, 2020. Date of publication February 5, 2020; date of current version October 23, 2020. This work was supported in part by the ARC Research Hub for Integrated Energy Storage Solutions and a UNSW Digital Grid Futures Institute grant. Paper no. TII-19-1002. *(Corresponding author: Abubakar Sadiq Sani.)*  Abubakar Sadiq Sani and Zhao Yang Dong are with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: sadiq.sani@ unsw.edu.au; joe.dong@unsw.edu.au).  Dong Yuan and Wei Bao are with the Faculty of Engineering, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: dong.yuan@ sydney.edu.au; wei.bao@sxsxstsydney.edu.au).  Color versions of one or more of the figures in this article are available online at [https://ieeexplore.ieee.org.](https://ieeexplore.ieee.org/)  Digital Object Identifier 10.1109/TII.2020.2971707 |

# I. INTRODUCTION

**E**

NERGY Internet is an intelligent smart grid that is integrated with Internet technologies to enable seamless interactionamongallenergyresourcesandachieveanincreasein energy efficiencies [1]. It optimizes operational schedules such as multiway communications and remote connect/disconnect of components by deploying an advanced metering infrastructure (AMI), which refers to an integrated system of technologies that enable bidirectional communication between service providers and their customers in the smart grid [2]. Fig. 1 presents a simple AMI architecture for the Energy Internet. This figure shows the bidirectional multiway communications between the components or users in the Energy Internet. The terms “bidirectional multiway communications” and “multiway communications” as well as “components” and “users” are, therefore, often used interchangeably. As multiway communication is an essential function in the AMI, the security of data is an increasingly critical concern. Additionally, as the Energy Internet expands, the AMI risks massive security issues due to cyberattacks such as side-channel attack (that gathers secret information from the implementation of cryptographic protocols such as key exchange) [3], false data injection attack (that compromises the AMI components during key exchange),replayattack,andman-in-the-middleattack,andlackof meeting the AMI security requirements such as confidentiality, integrity, availability, identification, authentication, and access control [4], [5]. Despite substantial security findings on AMI communications and remote connect/disconnect of components in the smart grid, a key exchange protocol (KEP) that deals with all the security requirements, mitigates side-channel and false-data injection attacks, secures multiway communications and remote connect/disconnect of components, and provides security in arbitrary adversarial environments (i.e., universal composition) has not been widely adopted for the AMI.



|  |
| --- |
| 1551-3203 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information. |

Researchers have identified the inabilities to meet many security and performance requirements of the AMI as the reasons for this limited adoption. Integrating the AMI security requirements has not been widely considered in most KEPs designed for securing communications. Besides, AMI performance requirements such as latency and reliability targets, which range from 2 to 15 s and from 99% to 99.99% [6], respectively, have not been widely treated. Recently, Mohammadali *et al.* [7]

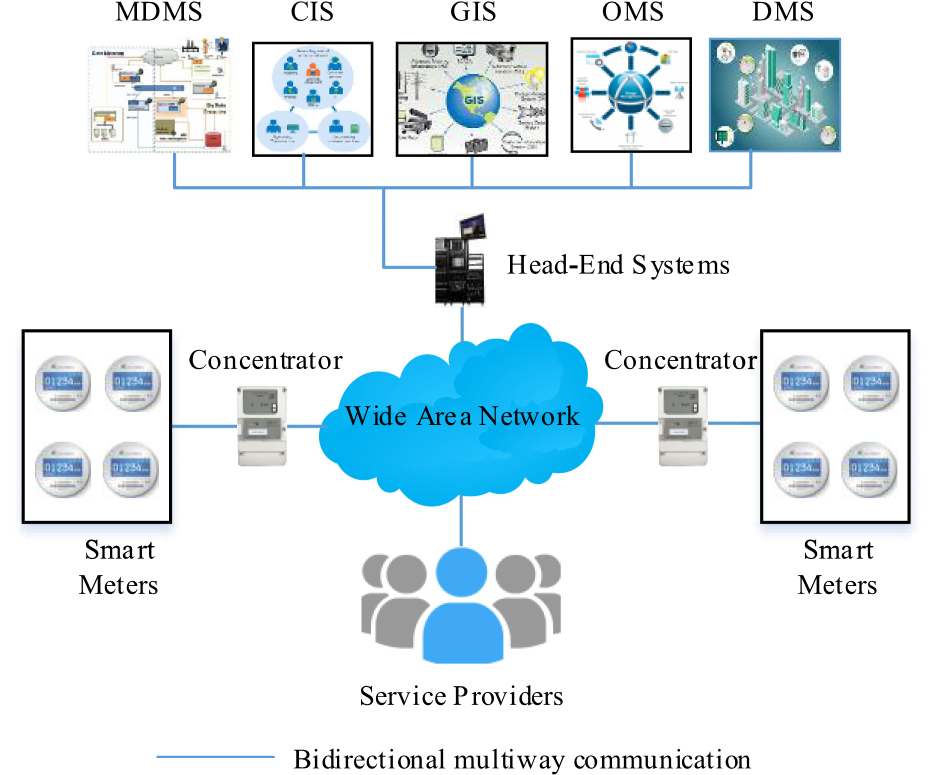


Fig. 1. Simple AMI Architecture. Abbreviations: MDMS—Meter Data Management System, CIS—Customer Information System, GIS— Geographic Information Systems, OMS—Outage Management Systems, DMS—Distribution Management System. The network connects smart meters to head-end systems, which manage communications between smart meters and MDMS, CIS, GIS, OMS, and DMS.

used an elliptic curve cryptographic (ECC) [8] algorithm, i.e., the elliptic curve Diffie–Hellman (ECDH) KEP to present two novel identity-based key establishment protocols, *NIKE* and *NIKE*+, for AMI in the smart grid. These protocols ensure integrity, availability, identification, and mutual authentication during key exchange, but leave open confidentiality and access control problems and do not provide universal composition.

One important goal for this article is, therefore, to provide a KEPthatmeetstheAMIsecurityandperformancerequirements, which we utilize for the Energy Internet. As the dangers cyber attacks pose to the smart grid increase, the Energy Internet will be faced with a choice of adopting a KEP with universal composition. Some KEPs have been proposed for the smart grid [7], [9]– [12]. However, to the best of our knowledge, there is no existing protocol that satisfies all the AMI security and performance requirements presented in this article and further provides universal composition. The existing KEPs face at least oneofthefollowingmajorchallenges:1)integratingalltheAMI security requirements is practically infeasible due to satisfying its performance requirements and incurring high computational and communication costs [9], [11]; 2) providing universal composition is difficult due to the commitment problem (i.e., once a shared secret session key has been created, neither  may become corrupted, where *dA* is a private key of userIDisapublickeyofuserID*B* [13])andthelackof built-in modularity [10], [11], [12]; and 3) inability to mitigate side-channel and false data injection attacks [7], [9]. Therefore, many of the existing KEPs for the AMI lack adequate security and performance feasibilities in the Energy Internet.

In this article, we present an AMI KEP based on universal composability [14], [15], which provides built-in modularity and security in arbitrary adversarial environments. Recall that, universal composability models are expressed by so-called ideal functionality. A real (key exchange) protocol *P* may use an ideal functionality *F* as a function to perform its cryptographic operations. Then, we can show that *P* (along with *F*) realizes another ideal functionality, say *F* . The universal composability composition theorems can now allow us to replace *F* by its realization, say *P*, which then implies that *P* using *P* realizes *F* . In our protocol, we provide an ideal crypto functionality *F*CR that supports a new authenticated and access control-based ECDH KEP, AA-ECDH, based on the decisional Diffie–Hellman (DDH) assumption [16], which refers to a computational hardness assumption used to prove the security of cryptographic protocols. The AA-ECDH is based on the ECDH KEP and it enforces authentication and access control during key exchange. We show that *F*CR can be realized under standard cryptographic assumptions. More specifically, our contributions are as follows.

1. We propose *F*CR for cryptographic operations related to our KEP. This functionality supports cryptographic primitives such as symmetric/asymmetric encryption, MACing, and our AA-ECDH. We show that *F*CR meets the AMIsecurityrequirements,providesallexpectedsecurity properties, supports secure multiway communication and remote connect/disconnect of components in the AMI, and is realizable under standard cryptographic assumptions. Furthermore, we propose and prove a realization *P*CR for*F*CR basedonthestandardcryptographicassumptions. As *F*CR supports many cryptographic operations, the proof of this realization requires several hybrid arguments.
2. We propose an ideal functionality *F*KEMA for AMI key exchange with mutual authentication and perfect forward secrecy. This functionality allows shared secret session key to be used in an ideal manner, provides an effective transition of key exchange states, supports key usage, and meetstheAMIsecurityrequirements.Abriefcomparison of *F*KEMA with other related ideal key exchange functionalities is provided in Section VI.
3. We provide KEP, which uses *F*CR and *F*KEMA and mitigates side-channel and false data injection attacks. Furthermore, we show that KEP meets the AMI security requirements to secure multiway communications and remote connect/disconnect of components in the AMI. We further verify the security of KEP using a symbolic verificationtool,automatedvalidationofinternetsecurity protocols and applications (AVISPA) [17].
4. We conduct some experiments on Tmote Sky nodes [18] to analyze the performance of KEP with the additional support of a network simulator-3 (NS-3) tool [19], and then show that KEP meets the AMI latency and reliability requirements. Detailed performance comparison of KEP with existing related protocols is presented in Section IX.
5. We illustrate the usefulness of our functionalities by analyzing the ECDH KEP and find some weaknesses in it. We show that using our functionalities, an enhancement of the protocol is a secure universally composable (AMI) KEP.

The remainder of this article is organized as follows. In Section II, we give a brief description of related works. In SectionIII,wepresentthepreliminariesofthisarticle.InSectionIV, wepresenttheidealcryptofunctionalityforcryptographicprimitives *F*CR. The realization *P*CR of *F*CR and the proof that *P*CR realizes *F*CR are presented in Section V. Our ideal key exchange functionality *F*KEMA is presented in Section VI. In Section VII, we present our KEP. In Section VIII, we provide the security verification of our protocol using AVISPA tool. In Section IX, we provide the performance analysis of our protocol as well as its comparison with the existing related protocols. The case study is performed in Section X. Finally, Section XI concludes this article.

# II. RELATED WORKS

Many KEPs have been proposed for the AMI and smart grid and each has tried to achieve specific security and performance requirements [7], [9]– [12], [20]–[27]. Mohammadali *et al.* [7] proposed *NIKE* and *NIKE*+ (cf., Section I) since the ECC incurs a less computational cost. The protocols provide mutual authentication during key agreement. However, they lack universal composition, do not provide confidentiality and access control, and remain vulnerable to side-channel and false data injectionattacks.Moreover,theprotocolsrelyonpresharedkeys during mutual authentication thereby presenting extra computational and communication and making their implementations very difficult and expensive in real-world environments. Nicanfar and Leung [9] proposed two KEPs, *PAKE* and *EPAK*, based on a symmetric-key algorithm and the ECDH KEP. The protocols provide security for key exchange in the smart grid but still lack universal composition and do not support access control. Besides, the protocols rely on preshared keys for mutual authentication. Nicanfar *et al.* [10] proposed an efficient key management protocol for securing communications in the smart grid. The protocol provides confidentiality, integrity, identification, availability, and mutual authentication. While the protocol is vulnerable to side-channel and false data injection attacks, it still incurs a large computational cost and, thus, is not fit for the AMI in the Energy Internet.

Wan *et al.* [11] proposed two scalable key management protocols, *SKM* and *SKM*+, using the combination of public-key cryptography and key tree technique. However, *SKM* does not provide confidentiality and access control and *SKM*+ does not provide access control during key exchange. Odelu *et al.* [12] proposed a key agreement scheme to reduce computation overheads for smart meters and service providers in the smart grid. However, the scheme is vulnerable to side-channel and false data injection attacks, and it incurs huge computational and communication costs. Moreover, the scheme does not provide confidentiality and access control.

Kewei *et al.* [20] designed a two-phase authentication and key agreement scheme to enable secure data readings from isolated smart grid devices. Abbasinezhad-Mood and Nikooghadam [21] identified a security flaw in [20]. To address the deficiency in [20] and provide secrecy during the key exchange, Abbasinezhad-Mood and Nikooghadam [21] presented an anonymous password-authenticated KEP to read isolated smart meters. However, the protocol [21] and scheme [20] incur large computational costs. Moreover, the protocol [21] does not provide confidentiality and access control and the scheme [20] does not provide access control.

Wu and Zhou [22] combined the symmetric-key and publickey techniques to present a fault-tolerant and scalable key management scheme that eliminates man-in-the-middle attack and replay attack. Xia and Wang [23] later showed that the scheme [22] is vulnerable to the man-the-in-middle attack. Furthermore, Park *et al.* [24] found the scheme [23] vulnerable to impersonation and unknown key-share attacks; however, the schemes [22]– [24] are vulnerable to side-channel and false data injectionattacks,anddonotprovideaccesscontrolanduniversal composition.

Seferian *et al.* [25] proposed an identity-based key distribution framework for the AMI. The framework provides security features for mitigating the side-channel attack. Tsai and Lo [26] combined an identity-based signature scheme and an identitybased encryption scheme to achieve security and efficiency for key distribution in the smart grid. They proposed a tamper-proof module for storing key data; however, the scheme is vulnerable to side-channel and false data injection attacks and does not provideconfidentialityandaccesscontrol.Kumar*etal.*[27]proposed a lightweight authentication and key agreement scheme, *LAKA*, for smart metering in the smart grid. However, the scheme assigns a secret token to every component before key agreement, thereby introducing extra computational and communication costs and secret token secrecy problem that can cause man-in-the-middle, side-channel, and false data injection attacks.

Although many of the proposed solutions succeeded in providing secure and efficient key exchange for the AMI, these solutions suffer from one or more of the following shortcomings: 1) lack of meeting all the AMI security requirements to support secure (multiway) communications and remote connect/disconnect of components [7], [9], [21]; 2) lack of mitigating side-channel attack and/or false-data injection attack [9]– [11]; 3) analysis of the solution is either based on symbolic methods and/or theoretical security analysis, which does not offer universal composition [7], [9]– [12], [20], [21], [26], [27]; 4) the preshared keys assumption, which relies heavily on another KEP that is not presented in the solutions [7], [9], [10]; 5) high computational and communication costs[20],[21],[26],[27];and6)lackofevidenceaboutmeeting the AMI latency and reliability requirements [7], [9], [10]. In this article, our AMI latency and reliability targets are 2 s and 99.99%, respectively. Table I presents a summary of functionalities and limitations of the related approaches. In this article, we propose a key exchange solution that addresses all the above shortcomings and more importantly, it provides mutual authentication without relying on another KEP for a preshared key computation (by an AMI service provider or trusted authority) and supports secure multiway communication and remote connect/disconnect of components. Note that we consider relying on the service provider (or trusted authority) for a preshared key as an AMI operational problem that brings new security threats to the AMI. Furthermore, our analysis of the ECDH KEP shows that the protocol is not a secure universally composable (AMI) KEP even though it is widely used for key exchange in

|  |
| --- |
| TABLE I |

∗− this symbol represents the functionality is not presented in the protocol.

the AMI. Thus, we use our functionalities to enhance and prove the security of the protocol.

# III. PRELIMINARIES

In this section, we provide a brief description of the general notion of universal composability, the universal composability model we use, and details of the key exchange in the AMI.

## A. General Notion of Universal Composability

Universal composability involves ideal and real protocols. The behavior and security properties of a protocol are represented by ideal protocol, which is also known as ideal functionality. The protocol to design and analyze represents the real protocol, which should be as secure as the ideal protocol, i.e., it should realize the ideal protocol. An adversary for the real protocol is called a real adversary, while an adversary for the ideal protocol is called an ideal adversary or a simulator. For every real adversary on the real protocol, there exists the ideal adversary on the ideal protocol.

## B. Inexhaustible Interactive Turing Machine (IITM) Model

The universal composability model we use is called the IITM model [14], [28] with responsive environments [29]. The IITM model is a system of an interactive turing machine with named bidirectional tapes. An interactive turing machine is a probabilistic polynomial-time turing machine with two named bidirectional tapes. In a system *Ss* = *M*1|*...*|*Mk*|!*M*1|*...*| of IITM model, where *M*1 and *Mk* are machines and ! indicates that an unbounded number of new copies of machines may be generated in a run of *Ss*, the connections of machines in *Ss* can be determined using their names. Two systems *Ss* and *St* are indistinguishable, i.e., *Ss* ≡ *St*, if the difference between the probabilities that *Ss* outputs 1 and *St* outputs 1 is negligible. We have three different types of systems in the IITM model. These systems include real and ideal protocols/functionalities, adversaries and simulators, and environments. The real and ideal protocols’ systems and the environment systems may have input/output (I/O) and network tapes or interfaces, while the adversarial systems only have a network tape. In this article, the so-called restricting messages [29] on the network are used to ensure that the environment and adversarial systems are always responsive, i.e., they always respond to requests or messages in the form (Respond*,id,m*), where *id* and *m* are random bit strings. The messages prevent the adversary from interfering with a protocol execution and ensure that the adversary can provide cryptographic values when requested by an ideal functionality. We now define simulation-based security in universal composability, which denotes strong simulatability.

*Definition 1:* [31]. Let *P* and *F* be protocol systems, i.e., the real and ideal protocol. respectively, with similar I/O interfaces. Then, *P* ≤ *F*, i.e., *P* realizes *F*, if there exists an adversarial system *S*, i.e., an ideal adversary or a simulator, such that *P* and *S*|*F* have similar external interfaces and for all environment systems *E*, connecting to the external interface of *P* and, thus, *S*|*F*, it holds that *E*|*P* ≡ *E*|*S*|*F*, where the adversary in *E*|*P* is subsumed by *E* using strong simulatability.

In the IITM model, there are several composition theorems [14], [15]; one of which allows a fixed number of the ideal and real protocols’ systems to be handled concurrently.

*Theorem 1:* [31]. Let *P*1 and *P*2 be real protocol systems and let *F*1 and *F*2 be ideal protocol systems such that *P*1 and *P*2, as well as *F*1 and *F*2 only connect with each other via their I/O interfaces and *Pi* ≤ *Fi*, for *i* ∈ {1*,*2}. Then, it holds true that *P*1|*P*2 ≤ *F*1|*F*2.

Othercompositiontheoremsin[14]and[15]canbecombined to construct more complex systems iteratively.

## C. Details of Key Exchange in the AMI

As more components are added to the AMI, secure multiway communications and remote connect/disconnect of components are required in the Energy Internet. For any secure multiway communication and remote connect/disconnect of components, thecomponentsarerequiredtoaccepteachotherbeforederiving a shared secret session key via a KEP, say *KEP*. In this article, we say that real-time access to a secure distributed database is granted to legitimate users. We assume that there is an AMI trusted authority that creates the database and legitimate users can securely search for information in the database. Brief description of side-channel and false data injection attacks is given as follows: 1) side-channel attack, i.e., a form of attack where an adversary gathers secret information from key exchange implementation and further recovers cryptographic keys, for example, smart meters are vulnerable to this attack, which leads to unauthorized access of customers’ data; and 2) false data injection attack, i.e., a form of backdoor attack where the adversary compromises components and, then, disrupts key exchange implementation, for example, falsified data are injected into smart meters, thereby manipulating the meters and affecting the smart grid. To mitigate side-channel and false data injection attacks during a key exchange in the AMI, we want to have a KEP that meets the AMI security requirements.

# IV. IDEAL CRYPTO FUNCTIONALITY FOR CRYPTOGRAPHIC PRIMITIVES IN AMI KEY EXCHANGE

We present our ideal crypto functionality *F*CR that supports our KEP in the AMI of the Energy Internet. In *F*CR, users can authenticateeachother,verifykeyexchangeaccess,andmeetthe AMI security requirements before deriving a shared secret session key. A KEP *P* can use *F*CR for its cryptographic operations. *P* using *F*CR realizes an ideal key exchange functionality *F*, i.e., *P*|*F*CR ≤ *F*. *F*CR guarantees that only authenticated users with key exchange access can exchange a shared secret session key. *F*CR allowsitsuserstoperformthefollowingkeyexchangecryptographic operations in an ideal manner: 1) generate an access control value for verifying key exchange access; 2) generate authenticated and access control-based private key, public key, AA-ECDH key, and shared secret session key; 3) verify the generated access control value and public key; 4) encrypt and decrypt messages using AA-ECDH key; and 5) compute and verify message authentication codes (MACs) using AA-ECDH key.

Formally, *F*CR is a machine with *n* I/O tapes and a network tape. The *n* I/O tapes represent different roles in a protocol while the network tape is used for communicating with the adversary. In runs of a system which has *F*CR, only one instance of *F*CR is available to handle all requests. *F*CR is parameterized with an elliptic curve domain parameters algorithm *EP*(1*η*) that is used to generate the domain parameters, where *η* is a security parameter. This algorithm takes *η* as input and returns (*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 the order of *G*, and *h* is a cofactor [8]. A user of *F*CR is identified by a tuple (ID*,*sid*,r*), where ID is the user identity, sid is a session identifier that is selected and managed by the protocol, and *r* is the role/tape connecting the user to *F*CR, ID, and sid. To ensure that *F*CR recognizes the user who sent/receives a message, all messages on the I/O tapes are prefixed with (ID*,*sid). Furthermore, *F*CR uses the restricting messages presented in Section III. If an environmentfailstorespondtosuchmessages,*F*CR repeatsthese messages until it receives an expected response, which is within an acceptable range of time. To provide security guarantees for secret keys, users do not get actual keys such as private keys and symmetric keys in *F*CR. Instead, they get pointers to such keys, which can be used to perform cryptographic operations. Before the cryptographic operations are performed using the pointers, the pointers are first replaced by the keys they refer to.

Upon the initialization of *F*CR for the first time, *F*CR executes *EP*(1*η*) and stores thegenerated (*p,a,b,G,n,h*). Furthermore, we expect *F*CR to receive a list of users’ identities with key exchange access from the AMI trusted authority that is known for accuracy and availability but not security and privacy and can always update information in its database. Then, *F*CR uses a restricting message to send the algorithm, list of users’ identities and key exchange access, and cryptographic algorithms request to the adversary. Upon the completion of this initialization, if a responseisreceivedonanetworktape,*F*CR returnscontroltothe adversary.Otherwise,*F*CR continuestoprocesstheoriginalmessage if the response is received on the I/O tape. The commands that *F*CR provides to a user are listed in Table II. Other commands such as Encryption using AA-ECDH key (Encryption), Decryption using AA-ECDH key (Decryption), MACing using AA-ECDH key (MACing), and VMACing using AA-ECDH key (VMACing) returns resulting ciphertext (say, *z*), plaintext (say, *y*), resulting MAC (say, *v*), and (*z,v*), respectively, to the user at the end of its execution, where *z* is a message. The description of the four aforementioned commands is omitted due to page limit. Note that 1) the Encryption command ensures *confidentiality*byencryptingdata;and2)theMACingcommand ensures *integrity* by applying a MAC to the data.

*Remarks:* 1) the environment or a user can request for the corruption status of any user or the user’s keys in *F*CR; 2) the use ofsidandaccesscontrolvalueAIsupportssecuremultiwaycommunication such that every communication path has aunique sid and AI; 3) to remotely connect or disconnect any customer in the AMI, the customer (say, with ID) and service provider (say, with ID) will generate a fresh AI for authentication and access control to support key exchange for secure remote connection or disconnection of the customer; 4) an AA-ECDH key is used as a preshared-keyduringkeyexchangetosolvetheAMIoperational problemofrelyingontheserviceprovider(oratrustedauthority) for a preshared-key; and 5) *F*CR supports *availability* since there is no existence of the AMI trusted authority between ID and IDduring the execution of all its commands. Furthermore, *F*CR can be easily extended to support multiple shared secret session keys derivation from a single AA-ECDH key and many access control values for multiway communication. However, we use the current formulation of *F*CR for simplicity and instead call the GenSeK command twice with  and  to obtain two keys.

# V. REALIZATION OF *F*CR

|  |
| --- |
| TABLE II    TABLE III |

In this section, we present a realization *P*CR of *F*CR. *P*CR implements all cryptographic operations of *F*CR via standard cryptographic schemes. It has the same network and I/O interfaces as *F*CR and parameterized with an algorithm *EP*(1*η*) with the samepropertiesasfor*F*CR,encryptionschemessuchasauthenc, unauthenc, pub for authenticated symmetric encryption, unauthenticated symmetric encryption, and public key encryption, respectively, a MAC scheme MAC, and two families of PRF *F* = (*Fη*)*η*∈*N* and *F* = (*Fη*)*η*∈*N* that take key(s) and salt as input,andoutputakey(formaldefinitionsofthesecryptographic primitives are provided in [31]). To initialize *P*CR, a message *m* is required to be sent. Then, *P*CR initializes itself by executing *EP*(1*η*) and storing the results before processing *m*. Similar to *F*CR, *P*CR maintains keys, access control values, and pointers to secret keys in a database within the functionality, and, then, uses them to decide the cryptographic primitive to execute. For example, *F* is used for deriving AA-ECDH keys from private keys and *F* is used for deriving session keys from AA-ECDH keys. Like in *F*CR, we say that: 1) all pointers are replaced by the keys they refer to before any related cryptographic operation is carried out in *P*CR; and 2) the environment can request whether any pointer or key of the user is corrupted. We describe how every command from *F*CR is implemented in *P*CR in Table III. Other commands such as Encryption, Decryption, MACing, and VMACing returns resulting ciphertext, plaintext, resulting MAC, and a message/resulting MAC value, respectively, to the user at the end of its execution. The description of the four aforementioned commands is omitted due to page limit.

We now prove that *P*CR realizes *F*CR using standard cryptographic assumptions. We say that the environment in *F*CR is well behaved if all cryptographic primitives fulfil the standard cryptographic assumptions. Specifically, we want to use the DDH assumption to prove that *P*CR ≤ *F*CR such that the simulator will provide an uncorrupted key when asked for an uncorrupted AA-ECDH key, and a corrupted key when asked for a corrupted AA-ECDH key, which cannot be inserted/used in *F*CR. Furthermore, we slightly restrict the environment not to cause the commitment problem [13] (cf., Section I) and key cycles [32] to capture the expected use of *P*CR*/F*CR in our KEP. We introduce a machine *M*+ that is placed between the environment and I/O interface of *P*CR*/F*CR to ensure that the environment satisfies our slight restriction and, thus, we say that the environment is well-behaved and used-order respecting. If an environment is no longer well-behaved or used-order respecting at any time, all future communications are stopped and blocked to prevent the violation of runtime condition. Note that corrupted values/keys are values/keys that cannot be verified in the database, thus these values/keys cannot be used for key exchange in *F*CR*/P*CR. We obtain the following theorem.

*Theorem 2:* Letauthenc,unauthencpub beencryption schemes, MAC be a MAC scheme, *EP* be an algorithm as above, *F* be a family pseudo-random functions, and *F* be a family pseudo-random functions for *EP*. Let *P*CR be parameterized with these algorithms. Let *F*CR be parameterized with *EP* Then,the following holds true:

*M*+|*P*CR ≤ *M*+|*F*CR (1)

if authenc is IND-CPA secure, unauthenc pub are INDCCA2secure,MAC isUF-CMAsecure,and*EP* alwaysoutput random form primes of field order *p* and groups with *n* ≥ 2 and such that the DDH assumption holds true for *EP*.

*Proof Sketch:* This proof consists of several hybrid systems as follows. In the first step, we define a hybrid system *P*CR1 where all asymmetric operations and access control value generation are handled and replaced by their ideal versions from *F*CR. In the second step, we define another hybrid system *P*CR2 where privatekeyhandlingisreplaced withitsidealversionandprivate key collisions and guessing are prevented. In the third step, we consider a hybrid system *P*CR3 where AA-ECDH key generation is replaced with its ideal version and the collision and guessing of any AA-ECDH key is not prevented. In the fourth step, a hybrid system *P*CR4 is presented where all symmetric operations and session key derivation are replaced with their ideal versions. In the fifth step, real MACs are replaced by their ideal versions. We conclude this proof by combining all the above steps and present that the simulator in this proof is responsive.

# VI. IDEAL FUNCTIONALITY FOR AMI KEY EXCHANGE

In this section, we present our ideal functionality *F*KEMA for AMI key exchange in the Energy Internet. It is based on mutual authentication and our AA-ECDH. Compared with other key exchange functionalities from [15], [30], and [31], *F*KEMA meets all the AMI key security requirements before a shared secret session key is derived and ensures the effective transition of key exchange states to control key exchange and usage in the AMI. Formally, *F*KEMA is a machine with two I/O tapes for key exchangeinitiatorandresponder,respectively,onenetworktape, and another two I/O tapes for connecting to *F*CR. It uses *F*CR as a functionforcryptographicoperations.Itmaintainskeyexchange states such as restricted, start, in-progress, finished, closed, and corrupted to effectively model the AMI security requirements and ensure key exchange and usage are successfully completed. Initially, all key exchange users are set to restricted before the start of a key exchange session.

We parameterized *F*KEMA with different types of symmetric keys *t*(key) such as preshared-key, authenc-key, unauthec-key, and MAC-key to help determine the type of key that is generated after a successful key exchange. A key exchange user in *F*KEMA is also identified by a tuple (ID*,*sid*,r*), where ID is a user identity, sid is a session identifier that is chosen and managed by the protocol, and *r* is a role of either a key exchange initiator or responder. Similar to *F*CR, every message from/to any I/O tapes is prefixed with (ID*,*sid). The operations provided by *F*KEMA to our KEP are as follows.

1. A user (ID*,*sid*,r*) with state restricted can start key exchange by sending a message *m* = (KEInit*,*, where IDis the user identity of the intended responder and *m*1 is a random bit string. Upon receiving this message, *F*KEMA verifies that ID and IDare recorded in the databaseandsetsIDastheresponderandstatesofIDand IDto started if the verification succeeds. The state started models *availability* as there is no existence of the trusted authority or any third party between the key exchange users.
2. Auser(ID*,*sid*,r*)withstatestartedcanuse*F*KEMA toaccess some commands of *F*CR. In this case, *F*KEMA gives the user access to the following commands: 1) GetSP; and 2) AccessI. Upon receiving an access control value AI as a response of *F*CR, this shows that the user has access to key exchange and *F*KEMA forwards this response to the user and keeps track of this value. Then, *F*KEMA sets the state of the user to in-progress. In the state started, *F*KEMA models *identification* (after the execution of the GetSP command) as well as *authentication* and *access control* (after the execution of the AccessI command).
3. A user (ID*,*sid*,r*) with state in-progress can use *F*KEMA to access some commands of *F*CR. *F*KEMA gives the user access to the following commands: 1) verify; 2) GenPrK; 3) GenEcK; 4) VMACing; 5) Encryption; 6) Decryption; 7) MACing; and 8) GenSeK. Upon receiving a pointer ptr asaresponseof*F*CR,thisshowsthatasessionkeyhasbeen successfully derived. Then, *F*KEMA forwards the response to the user and sets the state of the user to finished, which shows that the key exchange has been successfully completed. For multiway communication with two keys for two paths, the user is required to call the GenSeK command twice and once two pointers, ptr1 and ptr2, are received as a response, *F*KEMA forwards the response to the user and sets the state of the user to finished. In the state in-progress, *F*KEMA models *mutual authentication* (after the execution of the Verify command), *confidentiality* (after the execution of Encryption and Decryption), and *integrity* (after the execution of MACing and VMACing).
4. A user (ID, sid*,r*) with state finished can close key exchange by sending a messageKEClose*,*, where *m*2 is a random bit string. Upon receiving this message, *F*KEMA sets the state of the user to closed, notifies the adversary via a restricting message, returns Okay to the user after receiving a response from the adversary,

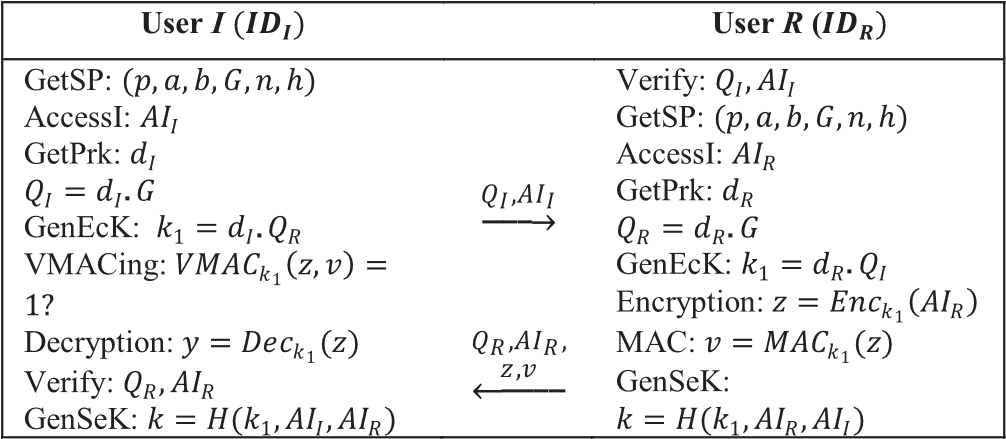


Fig. 2. Universally composable KEP.

and, then, the user loses access to all associated pointers or keys and has no access to *F*CR.

1. A user (ID*,*sid*,r*) with state closed can be corrupted by the adversary. Upon receiving this request, *F*KEMA updates the state of the user to corrupted.

*Remarks:* A user (ID*,*sid*,r*) with states restricted and closed can be corrupted by the adversary since the user does not have access to any commands, pointers, or session keys. Since the adversary does not have access to any keys or pointers after the key exchange session has been closed, *F*KEMA models perfect forward secrecy to ensure the security of session keys for secure multiway communications and remote connect/disconnect of components in the AMI. Thus, all session keys in *F*KEMA cannot be accessed by a corrupted user. We restrict any corruption of *F*KEMA to make it easier to use by our protocol since the session keys computed are used for a short period.

# VII. KEY EXCHANGE PROTOCOL

In this section, we present our KEP, *KEP*, as depicted in Fig. 2. It is based on the AA-ECDH and all the AMI security requirements. This protocol has important differences, which among others makes it more secure for the AMI. In particular, neither universal composition nor meeting all the AMI security requirements were considered in [7], [9]– [12], [20], [21], [26], [27].

We use two machines *MI*(*KEP*) and *MR*(*KEP*) to model the role of key exchange initiator and responder, respectively, in *KEP*. These machines have similar I/O tape as *F*KEMA and each has a network tape. *F*CR is used as a function for cryptographic operations by these machines and *KEP* realizes *F*KEMA. In every run of *KEP*, one instance of *MI*(*KEP*)*/MR*(*KEP*) per user (ID*,*sid) is available to execute it as shown in Fig. 2. At the end of *KEP*, the instances compute a session key. Then, a pointer to the session key is returned to the users and they can use it to perform ideal cryptographic operations.

The following theorem states that *KEP* is a secure universally composable AMI KEP.

*Theorem 3:* Let *MI*(*KEP*) and *MR*(*KEP*) be machines modeling the *KEP* as described above. Let *FCR* be the ideal crypto functionality for cryptographic primitives, and let *F*KEMA be the ideal functionality for AMI key exchange with parameter *t*(key) = authenc-key. Then, the following holds true:

*.* (2)

*Proof Sketch:* We say that a user ID is corrupted if the AA-ECDHkeyusedasapreshared-keyiscorrupted.Wesaythat an instance (ID*,*sid*,r*) is corrupted if the control of this instance is taken over by the adversary. We define a simulator *S* and show that *E*|*MI*(*KEP*)|*MR*(*KEP*)|*F*CR ≡ *E*|*S*|*F*KEMA for all responsive environments *E* ∈ *Env*(*MI*(*KEP*)|*MR*(*KEP*)|*F*CR). *S* simulates the protocol *MI*(*KEP*)|*MR*(*KEP*)|*F*CR, synchronizesthesimulatedinstancesof*MI*(*KEP*)|*MR*(*KEP*),andkeeps corrupted statuses of instances in *F*KEMA. To initialize *F*CR, *S* first sends a message to *F*CR and receives domain parameters (*p,a,b,G,n,h*) in response. Then, it asks *E* for related cryptographic algorithms. If *F*KEMA shows that a user (ID*,*sid*,r*) has started a key exchange, *S* does the same.

If an uncorrupted responder (ID*R,*sid*R,R*) accepts a public key *QI* = *dI.G* and access control value AI*I* and outputs a session key pointer, then *S* instructs *F*KEMA to compute a session from (ID*I,*sid*I,I*) and the instance (ID*R,*sid*R,R*) that created the preshared-key used in the second message of the protocol. Next, the function *F*CR of *F*KEMA will, then, request *S* to provide the session key value. The value is provided by *S*, which also uses it as a session key in its simulation. Finally, *F*KEMA is instructed by *S* to output the session key pointer for (ID*R,*sid*R,R*). Due to the use of restricting messages, we assumethatalloperationsperformedby*F*CR areuninterruptible. As *S* provides all uncorrupted keys, the environment *E* cannot insert any corrupted keys.

We show that the simulation is perfect in case of an uncorrupted instance of an initiator during the key exchange phase. Let (ID*I,*sid*I,I*) be an uncorrupted instance of *MI*(*KEP*) that wants to establish a key exchange session with a user ID. If (ID*I,*sid*I,I*) outputs a session key pointer, this shows that it must have accepted the second message of the protocol and the encryption/MAC key of its key exchange partner IDmust still be corrupted, else the protocol will stop since the environment is no longer well-behaved as presented in Section V.

Thus, we say that there exists some responder instance, say  that is uncorrupted. Furthermore, we say that the instance  cannot be corrupted by the adversary as the user ID is still uncorrupted.

Note that other cases such as an uncorrupted instance of a responder duringthekeyexchange phase, uncorrupted instances during the key usage phase, and corrupted instances are omitted due to page limit.

By Theorem 2, we can now replace *F*CR by *P*CR, which shows that *KEP* using actual cryptographic operations is a secure universally composable AMI KEP.

*Proposition 1:* Let *MI*(*KEP*) and *MR*(*KEP*) be machines as described above. Let *F*CR, *P*CR, and *M*+ be as defined in Theorem 2. Then, the following holds true:

*.* (3)

*Proof Sketch:* This proof follows easily from Theorem 1, Theorem 2, and Theorem 3 that *M*+|*MI*(*KEP*)|*MR*(*KEP*) creates a well-behaved environment when combined with any environment *E*. The well-behaved property cannot be violated by corrupted instances since they do not have access to uncorrupted keys. For uncorrupted instances that want to violate the wellbehaved property, we say that this does not occur since private keys cannot be accessed or reused after a session key has been created with these keys.

*Remarks: KEP* mitigates the side-channel and false-channel attacks. To see these: First, the combined use of *Enck*1(*.*) and MAC*k*1(*.*) prevents the leakage of secret information, which mitigates the side-channel attack. Note that: 1) encryption or MAC of *QI* and AI*I* is not required since both *QI* and AI*I* are not secret information; and 2) our protocol only focuses on the side-channel attack during key exchange and within the scope of this article. Finally, since *k*1 (derived via the GenEcK command), *Enck*1(*.*), and MAC*k*1(*.*) are utilized for securing the key exchange, false data injection attack is also mitigated as the messages *z* and *v* cannot be changed or altered because the adversary does not know the corresponding *k*1, which provides access to both users *I* and *R*. Thus, unlike many of the existing related KEPs in the related works (see Section II), *KEP* is resilient against side-channel and false data injection attacks. The shared secret session key derived from *KEP* can now be used for secure multiway communication and remote connect/disconnect of components.

# VIII. SECURITY VERIFICATION USING AVISPA TOOL

In this section, we simulate *KEP* using the widely-accepted formal verification tool, AVISPA [17], which is used for automatic security analysis of cryptographic protocols. The AVISPA tool has successfully been used for the security evaluation of AMI in the smart grid [7]. We use the high-level protocol specification language [33] for implementing *KEP* and follow theDolev–Yaoattackmodel[34](asitsadversarymodel)fordescribingtheknowledgeoftheadversary.IntheDolev–Yaoattack model,communicationbetweenusersisperformedoverapublic channel, where an adversary can intercept, eavesdrop, modify, inject, replay, and delete data being transmitted. We selected the widely-used on-the-fly model-checker (OFMC) [35] and constraint logic based attack searcher (CL-AtSe) [36] backends in AVISPA to check whether there are any attacks on *KEP*. We have simulated *KEP* using the Security Protocol ANimator (SPAN) for both OFMC and CL-AtSe to prove its security and authentication properties.

Without loss of generality, *KEP* is simulated in AVISPA without using pointers, i.e., keys are directly used, and the users do not exchange any data with the adversary. In our simulation, we modeled the following: 1) roles of the users in the key exchange states: 2) key exchange session; 3) composition of several instances of *KEP*; 4) intruder knowledge in *KEP*; and 5) secrecy and authentication goals in *KEP*. The simulation results from the OFMC and CL-AtSe backends shown in Fig. 3 confirm that *KEP* is safe against replay and man-in-the-middle attacks, and the symmetric key constructed from it is safe from the Dolev–Yao attack model.

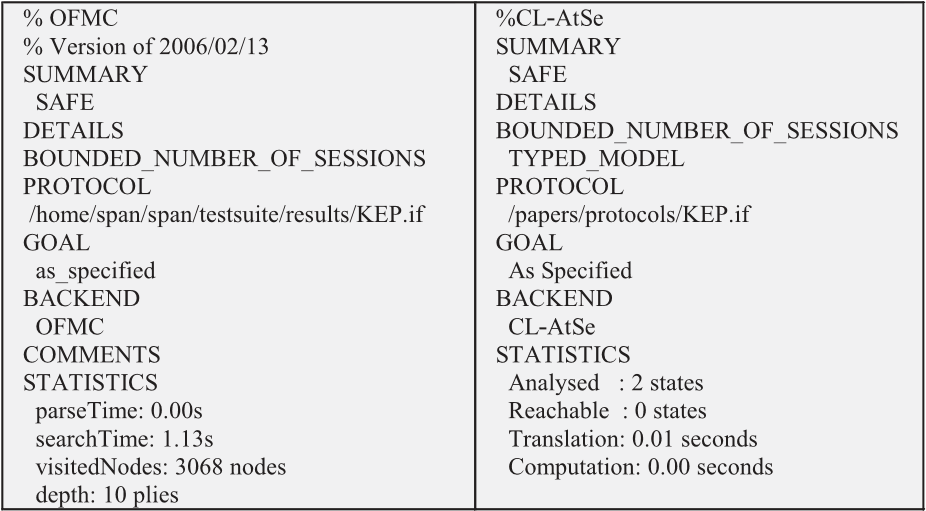
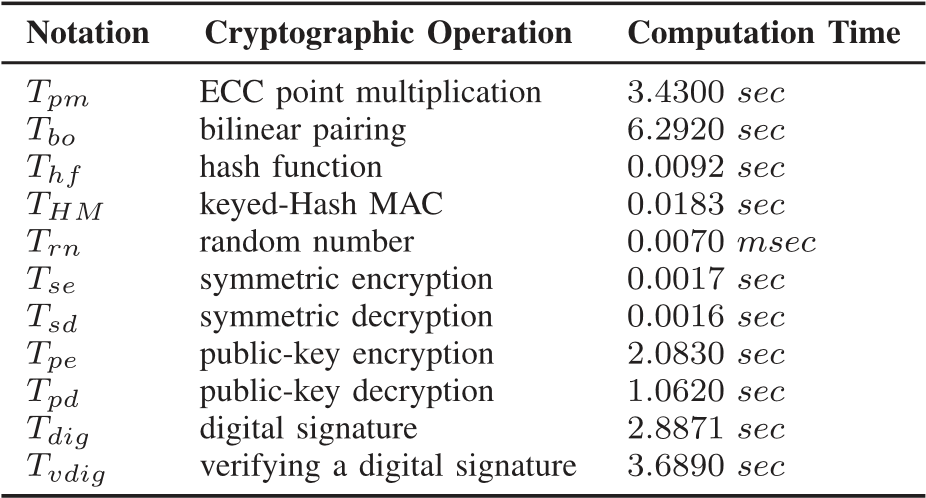


Fig. 3. Simulation results using OFMC and CL-AtSe backends.

TABLE IV

COMPUTATION TIMES OF CRYPTOGRAPHIC OPERATIONS



# IX. PERFORMANCE ANALYSIS

In this section, we analyze the performance of our protocol with the existing related protocols in terms of computational and communication costs, and then present the end-to-end delay (EED) and the reliability of our protocol. The notations of cryptographic operations are given as: 1) *T*pm—an ECC point multiplication;2)*T*hf—ahashfunction;3)*T*HM—aHMACfunction;4)*T*rn—arandomnumber;5)*T*se—asymmetricencryption; 6) *T*sd—a symmetric decryption; 7) *T*pe—a public-key encryption; 8) *T*pd—a public-key decryption; 9) *T*dig—a digital signature;10)*T*vdig—verifyingadigitalsignature;11)*Tbo*—abilinear pairing; and 12) *Teo*—a modular exponentiation. To realistically compare our protocol with the existing related protocols, we use Tmote Sky motes [18] to implement the protocols through the execution of their cryptographic operations, which are compiled under an operating system, TinyOS [37] and conducted on a MacBook Pro Machine (Intel Core i5-6500 CPU @ 3.20 GHz with 16 GB RAM). The computation times of the cryptographic operations are presented in Table IV. Note that the programs of the motes are written in nesC language [38]. We ignore the computation time of *Teo* without loss of generality and it does notimpactthecomputationalcostsoftheprotocolsinthisarticle.

## A. Computational Cost

This cost represents the number of cryptographic operations and the time required for the execution of key exchange. In Table V, we compare the computational cost of our protocol with the existing related protocols. The cost of every pointer corresponds to the cost of the key that it points to. In our

TABLE V COMPUTATIONAL COSTS COMPARISON



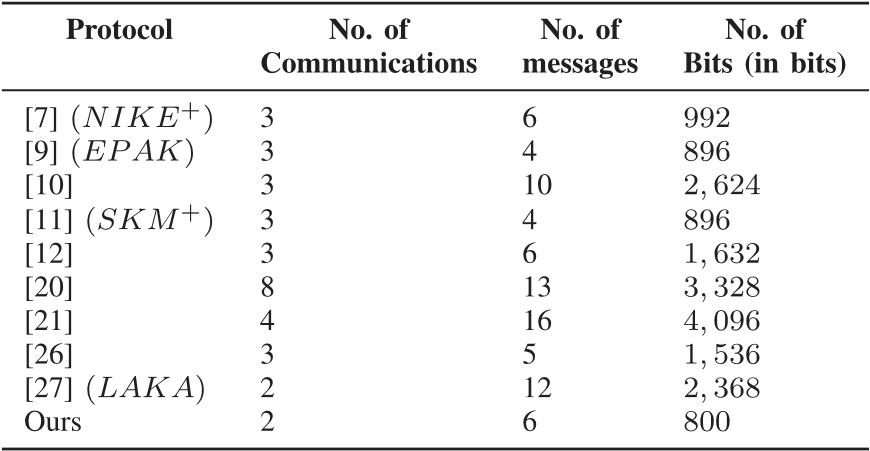
protocol, the computational costs required for users *I* and *R* are

2*T*pm + *T*rn + *T*HM + *T*sd + *T*hf ≈ 6*.*8891 s and 2*T*pm + *T*rn +

*T*HM + *T*se + *T*hf ≈ 6*.*8892 s, respectively. In this article, Kewei *et al.’s* protocol [20] has the longest computation time because of using many hash function operation. The least computational costs belong to Nicanfar and Leung’s protocol [9] (≈13*.*7634 s) andourprotocol(≈13*.*7783s),whichmeetsalltheAMIsecurity requirements and provides universal composition compared to Nicanfar and Leung’s protocol [9] and the other protocols.

Furthermore, the overall computational cost of our protocol as well as Nicanfar and Leung’s protocol [9] is efficient than the other protocols in [7], [10], [11], [12], [20], [21], [26], and [27] as the total number of cryptographic operations and computation time of our protocol and Nicanfar and Leung’s protocol [9] is 12*/*13*.*7783 and 14*/*13*.*7634 s, respectively, while it is 11*/*17*.*2144, 31*/*16*.*4386, 8*/*26*.*3041, 21*/*29*.*7341, 18*/*47*.*2144, 46*/*30*.*0564, 21*/*36*.*594, and 23*/*20*.*7426 s for the protocols in [7], [10], [11], [12], [20], [21], [26], and [27], respectively. Thus, Nicanfar and Leung’s protocol [9] and our protocol required the least computational cost due to the use of a smallnumberoflowcomputationtimecryptographicoperations

TABLE VI COMMUNICATION COSTS COMPARISON



such as *T*rn and *T*hf. It is very important to note that Nicanfar and Leung’s protocol [9] still incurs extra computational and communication costs from preshared keys (cf., Section II).

## B. Communication Cost

This cost represents the number of communications, and the number and size of messages to be transmitted during key exchange. Our protocol requires two communications and six messages. The communication costs of all the protocols are presented in Table VI. The number of communication bits in this tableisbasedonvariouslengthofbinarysequencessuchas:1)random number—32 b; 2) hash function—160 b; 3) MAC—256 b; 4) user identity—160 b; 5) symmetric encryption—128 b; 6) public key encryption—160 b; and 7) digital signature—160 b. The number of communication bits required in our protocol are given as: 1) |*QI,*AI*I*| = 192 b; and 2) |*QR,*AI*R,z,v*| = 608 b. Thus, the total number of communication bits required in our protocol is 800 b. The least communication cost belongs to our protocol, while the highest communication cost belongs to Abbasinezhad-Mood and Nikooghadam’s protocol [21] due to the use of many messages with large amount of communication bits. Compared to Nicanfar and Leung’s protocol [9] and Wan *et al.*’s protocol [11], which have a total number of four messages each, our protocol still offers the least number of communication bits due to the use of messages with small amount of communication bits such as AI*I* and AI*R*.

## C. End-to-End Delay

In this section, we use a widely accepted simulation tool, NS-3[19]tomeasuretheEEDof*KEP*.TheEEDisrepresented as the average time at which the key exchange data packets arrive at the destination from the sender. We measure the EED to show the impact of *KEP* on the AMI. The key details and descriptions of the parameters used in NS-3 are: 1) platform— Ubuntu16.04TLS;2)numberofusers—two,i.e.,users*I* and*R*; 3) mobility—100 Mb/s; 4) communication medium—Wi-Fi; 5) channel model—P2P; 6) routing protocol—OLSR; and 7) communication range—100m. The sizes of messages are 192 b for *I* → *R* and another 608 b for *R* → *I*. Other standard parameters are used from the NS-3 to support our evaluation. The maximum EED of *KEP* is 0*.*0273 s. Thus, the overall EED of *KEP* is below our 2 s AMI latency target. As a result, *KEP* is efficient for the AMI in the Energy Internet.

|  |
| --- |
|  |
|  |

Fig. 4. UC-based ECDH KEP.

## D. Reliability

In this article, the reliability represents the probability that the data packets of *KEP* are successfully transferred within a certaindelay.Wesaythatthereliabilityisaffectedifeitheradata packet is lost or a data packet is delivered too late. Our reliability target for the AMI is 99.99% with a latency of 2 s. In our NS-3 simulation,weobservedthefollowing:1)thetotalnumberofbits and packets sent by the sender is equal to the total number of bits and packets received by the receiver, respectively; 2) no packet was lost during transmission; and 3) the latency target of 2 s is satisfied.Thus,thereliabilityof*KEP* is≥99.99%within*<* 2s, which meets/exceeds the above target. As a result, *KEP* is fit for AMI of the Energy Internet, which shows that the success probability of transmitting data packets during secure multiway communications and remote connect/disconnect of components in the AMI is ≥99.99%.

# X. CASE STUDY

In this section, we carry out a case study to demonstrate the usefulness of our functionalities in a widely used KEP in the AMI. We analyze the ECDH KEP, which allows two users to establishasharedsecretkey.ManypracticalKEPssuchasANSI X9.63[39]supporttheECDHKEP.InFig.4,weshowtheECDH KEPinoursetting.Wemodeltheprotocolsimilarlyasthe*KEP*, except that we have new machines *MI*(ECDH) and *MR*(ECDH) executing the protocol in Fig. 4 to exchange a shared secret key *k*1. We argue that the ECDH KEP does not realize *F*KEMA and is vulnerable to active man-in-the-middle attacks. To prove this, we consider the following setting: an honest originator instance outputs a shared secret key that was generated from its private key *dI* and the responder’s public key *QR*. Consider a scenario where it is very possible that the responder might have received a different public key, say , in the first message of the protocol. If *QA* was not generated by *F*CR, then *dR* of *QR* will be marked as corrupted after the computation of *QA.dR* because there is no guarantee from the DDH assumption that the attacker does not learn anything about *dR* in this scenario. As *dR* is marked as corrupted, all keys derived from it will be markedascorrupted.Thus,wehavenosecurityguaranteefor*QI* and *QR* and an attacker can easily let the initiator and responder instances to accept any public key. This attack is referred to as the active man-in-the-middle attack, which is a direct attack on the ECDH KEP.

To fix this problem, we present an enhancement of the ECDH KEP. We need to have a value of user *I*, say *AII*, sent along with the user *QI*, where (*AII,QI*) is jointly available in a database, and also MACed the response *QR*, where a MAC value is computed using the MACing command in *F*CR. More precisely, we need to use *F*CR and *F*KEMA up to the point where a secret key is returned to the user in *F*CR, and *F*KEMA closed the key exchange session. This allows us to analyze the protocol using the DDH assumption, as now (AI*I,QI*) guarantees that the responder paired *QR* with *QI* only, and *QR/MAC*(*QR*) guarantees that theinitiatorpaired*QI* with*QR*.Wefurtherenhance theprotocol for the AMI by meeting the AMI security requirements since it has been widely used to support key exchange in the AMI and it lacks many of the AMI security requirements such as authentication, access control, confidentiality, and integrity. In thiscase,weequippedtheprotocolwithall*F*CR’scommandsand key exchange states in *F*KEMA to meet the security requirements andfurthermitigateside-channelandfalsedatainjectionattacks. Thus, the enhanced protocol can be used directly to derive presharedkeysandsharedsecretsessionkeysforsecuringmultiway communicationsandremoteconnect/disconnectofcomponents. We omit a theorem of the enhanced ECDH KEP due to page limit.

In the enhanced ECDH KEP, the computational cost required for user *I* (ID*I*) and user *R*(ID*R*) is 2*T*pm + *T*rn + *T*HM + *T*sd + *T*hf ≈ 6*.*8891 s and 2*T*pm + *T*rn + *T*HM + *T*se + *T*hf ≈ 6*.*8892 s, respectively. We can see that the protocol requires the same computational cost as the *KEP* (cf., Section IX-A) and, thus, theprotocolisalsoeffectivebasedonthe*KEP*’scomputational cost.

# XI. CONCLUSION

In this article, we identified security and performance challenges in existing related KEPs for the AMI and smart grid. Using a universal composability model, we proposed an ideal crypto functionality *F*CR that supports cryptographic primitives such as authentication and our new AA-ECDH and models the AMI security requirements such as confidentiality, integrity, availability, identification, authentication, and access control.

We also proposed an ideal functionality  for AMI key exchange, which supports the properties of *F*CR. Furthermore, we presented a universally composable AMI KEP, *KEP*, to address the challenges of key exchange for secure multiway communications and remote connect/disconnect of components in the AMI of the Energy Internet. Using *F*CR and *F*KEMA, we showed that our protocol guarantees strong universal composable security guarantees. We further utilized the AVISPA tool to complement the security evaluation of our protocol. We measured the performance of our protocol using theoretical analysis and the NS-3 simulator tool. The performance results showed that our protocol is more efficient in computation and communication when compared with many existing related protocols, and it meets the AMI performance requirements such as latency and reliability. Furthermore, we showed the usefulness of our functionalities in a case study, where we analyzed and enhanced the ECDH KEP that is widely used in the AMI. In future work, we plan to extend our functionalities to include accountability and auditing and manage secure energy demand and consumption for the Energy Internet.

# REFERENCES

1. Z.Y.Dong,“Towardsanintelligentfutureenergygrid.”[Online].Avail[able: http://www2.pv.unsw.edu.au/videos/Joe-Dong-15September2016/ slides/From-Smart-Grid-to-Energy-Internet-20160915\_public.pdf](http://www2.pv.unsw.edu.au/videos/Joe-Dong-15September2016/slides/From-Smart-Grid-to-Energy-Internet-20160915_public.pdf)
2. NETLModernGridStrategy,“Advancedmeteringinfrastructure,”Office Electr.EnergyRel.,USDept.Energy,2008.
3. Z. A. Baig and A.-R. Amoudi, “An analysis of smart grid attacks and countermeasures,” *J. Commun.*,vol.8,no.8,pp.473–479,2013.
4. V.Y.PillitteriandT.L.Brewer,“Guidelinesforsmartgridcybersecurity (No.NISTInteragency/InternalRep.(NISTIR)-7628Rev1),”2014.
5. B.Brown *et al.*,“AMIsystemsecurityrequirements,”AMISecurityTask Force,2008.
6. U.S.DepartmentofEnergy,“CommunicationsrequirementsofSmartGrid technologies,”USDept.Energy,Tech.Rep.,pp.1–69,2010.
7. A.Mohammadali,M.S.Haghighi,M.H.Tadayon,andA.MohammadiNodooshan, “A novel identity-based key establishment method for advancedmeteringinfrastructureinsmartgrid,” *IEEE Trans. Smart Grid*, vol.9,no.4,pp.2834–2842,Jul.2018.
8. A.H.Koblitza,N.Koblitzb,andA.Menezes,“Ellipticcurvecryptography: Theserpentinecorseofaparadigmshift,” *J. Number Theory*,vol.131, no.5,pp.781–814,2011.
9. H. Nicanfar and V. C. M. Leung, “Multilayer consensus ECC-based passwordauthenticatedkey-exchange(MCEPAK)protocolforsmartgrid system,” *IEEE Trans. Smart Grid*,vol.4,no.1,pp.253–264,Mar.2013.
10. H.Nicanfar,P.Jokar,K.Beznosov,andV.C.M.Leung,“Efficientauthenticationandkeymanagementmechanismsforsmartgridcommunications,” *IEEE Syst. J.*,vol.8,no.2,pp.629–640,Jun.2014.
11. Z.Wan,G.Wang,Y.Yang,andS.Shi,“SKM:Scalablekeymanagement for advanced metering infrastructure in smart grids,” *IEEE Trans. Ind. Electron.*,vol.61,no.12,pp.7055–7066,Dec.2014.
12. V.Odelu,A.K.Das,M.Wazid,andM.Conti,“Provablysecureauthenticatedkeyagreementschemeforsmartgrid,” *IEEE Trans. Smart Grid*, vol.9,no.3,pp.1900–1910,May2018.
13. R.CanettiandM.Fischlin,*UniversallyComposableCommitments*.Berlin, Germany:Springer,2001,pp.19–40.
14. R.Küsters,“Simulation-basedsecuritywithinexhaustibleinteractiveturingmachines,”in *Proc. 19th IEEE Comput. Security Found. Workshop*, 2006,pp.309–320.
15. R. Küsters and M. Tuengerthal, “Composition theorems without preestablishedsessionidentifiers,”in *Proc. 18th ACM Conf. Comput. Commun. Security*,Chicago,IL,USA,2011,pp.41–50.
16. M. Abdalla, M. Bellare, and P. Rogaway, “The oracle Diffie-Hellman assumptionsandananalysisofDHIES,”in *Proc. Cryptographers’ Track RSA Conf.*,2001,pp.143–158.
17. AVISPA.“Automationvalidationofinternetsecurityprotocolsandapplications.”[Online].Available:<http://www.avispa-project.org/>
18. TmoteSky,“UltralowpowerIEEE802.15.4compliantwirelesssensor module,”MoteivCorporation,Berkeley,CA,USA,2006.
19. [NS-3Consortium.“Thenetworksimulator3.”[Online].Available:https: //www.nsnam.org/](https://www.nsnam.org/)
20. S.Kewei,N.Alatrash,andW.Zhiwei,“Asecureandefficientframework toreadisolatedsmartgriddevices,” *IEEE Trans. Smart Grid*,vol.8,no.6, pp.2519–2531,Nov.2017.
21. D. Abbasinezhad-Mood and M. Nikooghadam, “Efficient anonymous password-authenticated key exchange protocol to read isolated smart metersbyutilizationofextendedchebyshevchaoticmaps,” *IEEE Trans. Ind. Informat.*,vol.14,no.11,pp.4815–4828,Nov.2018.
22. D.WuandC.Zhou,“Fault-tolerantandscalablekeymanagementforsmart grid,” *IEEE Trans. Smart Grid*,vol.2,no.2,pp.375–381,Jun.2011.
23. J.XiaandY.Wang,“Securekeydistributionforthesmartgrid,” *IEEE Trans. Smart Grid*,vol.3,no.3,pp.1437–1443,Sep.2012.
24. J.H.Park,M.Kim,andD.Kwon,“Securityweaknessinthesmartgrid keydistributionschemeproposedbyXiaandWang,” *IEEE Trans. Smart Grid*,vol.4,no.3,pp.1613–1614,Sep.2013.
25. V. Seferian, R. Kanj, A. Chehab, and A. Kayssi, “Identity based key distribution framework for link layer security of AMI networks,” *IEEE Trans. Smart Grid*,vol.9,no.4,pp.3166–3179,Jul.2018.
26. J.-L.TsaiandN.-W.Lo,“Secureanonymouskeydistributionschemefor smartgrid,”*IEEETrans.SmartGrid*,vol.7,no.2,pp.906–914,Mar.2016.
27. P. Kumar, A. Gurtov, M. Sain, A. Martin, and P. H. Ha, “Lightweight authenticationandkeyagreementforsmartmeteringinsmartenergynetworks,” *IEEE Trans. Smart Grid*,vol.10,no.4,pp.4349–4359,Jul.2019.
28. R.KüstersandM.Tuengerthal,“TheIITMmodel:Asimpleandexpressivemodelforuniversalcomposability,” *IACR Cryptol. EPrint Archive*, vol.2013,2013,Art.no.25.
29. J. Camenisch *et al.*, “Universal composition with responsive environments,”in *Proc. II, 22nd Int. Conf. Adv. Cryptol.*,2016,vol.10032.
30. R.KüstersandM.Tuengerthal,“Idealkeyderivationandencryptionin simulation-basedsecurity,”in *Proc. 11th Int. Conf. Topics Cryptol., CTRSA*,SanFrancisco,CA,USA,2011.
31. R. Küsters and D. Rausch, “A framework for universally composable Diffie-Hellmankeyexchange,”in *Proc. IEEE Symp. Secur. Privacy*,2017, pp.881–900.
32. J.Black,P.Rogaway,andT.Shrimpton,“Encryption-schemesecurityin thepresenceofkey-dependentmessages,”in*Proc.9thAnnu.Int.Workshop Sel. Areas Cryptogr.*,2003,pp.62–75.
33. D.vonOheimb,“Thehigh-levelprotocolspecificationlanguageHLPSL developedintheEUprojectAVISPA,”in *Proc. APPSEM Workshop*,2005, pp.1–17.
34. D.DolevandA.Yao,“Onthesecur.ofpublickeyprotocols,” *IEEE Trans. Inf. Theory*,vol.29,no.2,pp.198–208,Mar.1983.
35. D. Basin, S. Mödersheim, and L. Vigan , “OFMC: A symbolic model checkerforsecur.protocols,” *Int. J. Inf. Secur.*,vol.4,no.3,pp.181–208, 2005.
36. M.Turuani,“TheCL-Atseprotocolanalyser,”in*Proc.Int.Conf.Rewriting Techn. Appl.*,2006,pp.277–286.
37. J.Hill,R.Szewczyk,A.Woo,S.Hollar,D.Culler,andK.Pister,“System architecturedirectionsfornetworkedsensors,” *ACM SIGOPS Operating Syst. Rev.*,vol.34,no.5,pp.93–104,2000.
38. D.Gay,P.Levis,R.VonBehren,M.Welsh,E.Brewer,andD.Culler,“The *nesC*language:Aholisticapproachtonetworkedembeddedsystems,”*Acm Sigplan Notices*,vol.49,no.4,pp.41–51,2014.
39. ANSI,“X9.63:Publickeycryptographyforthefinancialservicesindustry,keyagreementandkeytransportusingellipticcurvecryptography,” AmericanNationalStandardsInstitute,2001.