Unbalancing Pairing-Free Identity-Based Authenticated Key Exchange Protocols For Disaster Scenarios

Jie Zhang, Xin Huang, Wei Wang and Yong Yue

***Abstract*—In disaster scenarios, such as an area after a terrorist attack, security is a significant problem since communications involve information for the rescue officers, such as polices, militaries, emergency medical technicians and the survivors. Such information is critically important for the rescue organizations; and protecting the privacy of the survivors is required. Normally, authenticated key exchange (AKE) is an underlying approach for security. However, available AKE protocols are either inconvenient or infeasible in disaster areas due to the very nature of disasters.**

To address the security problem in disaster scenarios, we propose two pairing-free identity-based AKE protocols that have unbalanced computational requirements on the two parties. Compared with existing AKE protocols, the proposed protocols have a number of advantages in disaster scenarios: 1) They are more convenient than symmetric cryptography-based AKE protocols since they do not require any pre-shared secret between the parties; 2) They are more feasible than asymmetric cryptography-based AKE protocols since they do not require any online server; 3) They are more friendly to battery-powered and computationally limited devices than pairing-based and pairing-free identity-based AKE protocols since they do not involve any bilinear pairing (a time-consuming operation), and have lower computational requirement on the limited party. Security of the proposed protocols are analyzed in detail; and prototypes of them are implemented to evaluate the performance. We also illustrate the application of the protocols through a vivid use case in a terrorist attack scenario.

***Index Terms*—authenticated key exchange, ID-based cryptography, pairing-free, unbalanced computational requirements, disaster scenarios.**

I. Introduction

THE past decades have witnessed a number of mass casualty disasters including both natural and human-made hazards, such as the Great East Japan Earthquake in 2011, the Indian Tsunami in 2004 and the 9/11 terrorist attacks in 2011. Fortunately, advanced information technology has saved numerous survivors by enabling collaborative work of different teams and organizations through handheld or wearable devices such as tablets, smart phones, medical devices and life detection instruments. By now, a number of schemes, systems and standards employing information technology in disaster scenarios are proposed, such as the IEEE 1512 Family of Standards [1][2][3][4].

A. Problem

When applying information technology in disaster areas, a significant problem is security. Data transmitted among rescue officers' devices carry information for their organizations; and such information is critically important by its nature [5]. If the information of the rescue organizations is leaked to illegal parties or criminals who sneak into the affected areas, disorganization will occur within these organizations [5]. Besides, the law requires that privacy of patients and victims should be protected [6]; and this also accords with ethics.

The very nature of diasters poses critical challenges to security. Since the network infrastructure is destroyed by the attack, the Internet is unavailable; and devices are connected by infrastructure-less manners [7][8][9]. In addition, devices brought into the affected area have to work for hours powered only by batteries due to the destruction of power infrastructure. These limitations make a number of available security schemes inconvenient or infeasible in disaster areas.

B. Motivation

Authenticated key exchange (AKE) is the foundation of security. It establishes authenticated session keys between two (or more) entities. For disaster scenarios, the most suitable AKE protocols are identity-based AKE (ID-AKE) protocols [10][11][12][13], in particular, pairing-free ID-AKE protocols [14][15][16] which are more lightweight than those involve bilinear pairings. They accord with the situation of disaster scenarios in the following aspects. First, they do not require any pre-shared secret. In disaster areas, it is inconvenient for devices to pre-share secrets since they may belong to different organizations. Second, they do not require online trusted servers. In disaster areas, it is often infeasible to access online servers since the network infrastructure may be destroyed.

This work was supported by the XJTLU research development fund projects under Grant RDF140243 and Grant RDF150246, in part by the National Natural Science Foundation of China under Grant No. 61701418, in part by Innovation Projects of The Next Generation Internet Technology under Grant NGII20170301, in part by the Suzhou Virtual Reality Lab Platform Project under Grant RRSP1012017029, in part by the Suzhou Science and Technology Development Plan under Grant SYG201516, and in part by the Jiangsu Province National Science Foundation under Grant BK20150376.

J. Zhang, X. Huang, W. Wang and Y. Yue are with Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, China. e-mail: jie.zhang03@xjtlu.edu.cn, xin.huang@xjtlu.edu.cn, wei.wang02@xjtlu.edu.cn, yong.yue@xjtlu.edu.cn

J. Zhang and W. Wang are also with Department of Computer Science, University of Liverpool, UK.

However, existing pairing-free ID-AKE protocols require both parties execute equivalent computational tasks. In disaster scenarios, communications often take place between a computationally-limited device and a powerful one. Moreover, devices in disaster areas often need to continuously work for hours powered merely by their batteries. Therefore, to better address the security problem in disaster scenarios, it is significant to reduce the computational requirement on the limited device in pairing-free ID-AKE protocols.

C. Contributions

We propose two pairing-free ID-AKE protocols that have unbalanced computational requirements on the two parties. In particular, we transfer scalar multiplications from one party to another in pairing-free ID-AKE protocols. The protocols are named Protocol I and II. Protocol I transfers one scalar multiplication from the limited initiating device to the more powerful responding device. Protocol II transfers one scalar multiplication in the opposite direction; it is used for scenarios where the initiating device is more powerful than the responding device.

The main contributions of our work include:

- A new idea is proposed to unbalance the computational tasks of pairing-free ID-AKE protocols. The idea can be applied to other existing paring-free ID-AKE protocols [15].
- Design of the unbalanced and pairing-free ID-AKE protocols. The unbalanced and pairing-free ID-AKE protocols establish authenticated session keys for two parties that one of them is a limited device. It dramatically reduces the computational burden on the limited device. Therefore, compared with existing AKE protocols, it has a significant advantage in addressing the security problem in disaster scenarios.

D. Paper Organization

The rest of this paper is organized as follows. In Section II, we review the related work. In Section III, we introduce the underlying cryptographic knowledge. In Section IV, we present the unbalanced and pairing-free ID-AKE protocols. In Section V, we provide security analysis of the proposed protocols. In Section VI, we implement prototypes of the proposed protocols and study the performance. In Section VII, we illustrate the application of the proposed protocols in disaster scenarios through a use case. Finally in Section VIII, we conclude this paper and present our future work.

II. Related Work

This section reviews typical AKE protocols in related work.

A. Symmetric-AKE Protocols

Symmetric cryptography-based AKE (denoted by "Symmetric-AKE") protocols [17][18][19] require the parties to share master keys in advance; or each entity shares a master key with the same online trusted server. Such protocols are inconvenient in disaster areas for the following reasons. First, rescue officers arrive at the disaster area at different times, and so do their devices. Therefore, it is inconvenient for a device that has already entered the affected area to pre-share master keys with all newly arrived devices. Second, the online trusted server is unaccessible since network infrastructure is destroyed by the very nature of the disaster.

B. Asymmetric-AKE Protocols

Unlike Symmetric-AKE, asymmetric cryptography-based AKE (denoted by "Asymmetric-AKE") protocols [20], [21] do not require pre-shared secrets. They allow parties to establish authenticated session keys from their public information (i.e., public keys) and ephemeral secrets. However, to guarantee the authenticity of the public keys, traditional asymmetric cryptography requires a public key infrastructure (PKI) [22] to maintain the certificates of the parties' public keys. The core component of PKI is an online certification authority (CA) that issues certificates of public keys. However, the unavailability of network infrastructure makes it impossible to implement the PKI in disaster areas.

C. ID-AKE Protocols

Identity-based (ID-based) cryptography [23] is a promising branch of asymmetric cryptography. It does not require any pre-shared secret or PKI. Since Boneh and Franklin [24] introduced the first ID-based encryption scheme from bilinear pairings, a number of ID-AKE protocols from bilinear pairings are proposed. Smart proposes the first ID-AKE protocol from Weil pairings in [10]. However, this protocol does not provide full forward secrecy [11]. To overcome the weakness of Smart's protocol, Shim proposes a new ID-AKE protocol from Weil pairings in [11]. In [12], Chen and Kudla review ID-AKE protocols and propose a new ID-AKE protocol which has provable security in the modified Bellare-Rogaway (mBR) model [25]. Scott presents unbalanced and paring-based ID-AKE protocols in [13].

However, all of the above protocols involve bilinear pairings which is a time-consuming operation. In order to alleviate the computational burden, pairing-free ID-AKE protocols are proposed. Cao *et al.* present a paring-free ID-AKE protocol with only two messages in [14]. Ni *et al.* present a pairing-free ID-AKE protocol [15] which has provable security in the extended Canetti-Krawczyk (eCK) model [26]. Recently, L. Dang *et al.* propose an efficient and pairing-free ID-AKE protocol with provable security [16].

Although the pairing-free protocols are more efficient than those involve bilinear pairings, they require equivalent computational requirements on the two parties. Therefore, they are not perfectly suitable for communications between a limited device and a powerful one in disaster areas.

D. Summary

We compare the aforementioned AKE protocols in Table I. According to the table, we have two findings: (1) Pairing-free ID-AKE protocols are the most suitable AKE schemes

TABLE II
SYMBOLS.

Symbol	Meaning
E	The elliptic curve.
G	The elliptic curve group.
F_p	The finite field with the prime order p.
$\mathcal{O}$	The point at infinity.
P	The generator of G.
P_x, P_y	The x and y coordinates of the point P.
$\times$	The scalar multiplication.
$\|\|$	The concatenation of bit strings.
ID	Identity.
$\mathcal{E}$	The attacker.
$Advantage^{\mathcal{E}}(k)$	The advantage of $\mathcal{E}$ under security parameter k.
Z^+	The set of positive integers.
H_1, H_2	Cryptographic secure hash functions.
$\mathcal{C}$	Cost.

for disaster scenarios since they do not require any pre-shared secret or PKI; and they are much lightweight than pairing-based ID-AKE protocols. (2) All these available AKE protocols require equal computations on the two parties even if they have different computational capabilities.

III. PRELIMINARIES

This section introduces the underlying cryptographic knowledge. In addition, we list symbols used in the paper in Table II.

A. Elliptic Curve Group

1) Definition:

Definition 1 (Elliptic Curve Group): An elliptic curve E over a prime finite field F_p with order p is defined by the following equation:

$$y^2 = x^3+\alpha x+\beta \mod p, \text{ where } \alpha, \beta \in F_p,\ 4\alpha^3+27\beta^2 \neq 0$$

The corresponding elliptic curve group G consists of all solutions (x, y) of the above equation and $\mathcal{O}$ which is the point at infinity. That is,

$$G = \{(x, y)|x, y \in F_p, (x, y) \in E\} \cup \{\mathcal{O}\}.$$

2) Operations: Let P and Q be points over G and n be an integer. Two types of operations over G are defined as follows.

- Point addition $+$. The point addition of P and Q is denoted by $P + Q$. The result is also a point over G.
- Scalar multiplication $\times$. The scalar multiplication between n and P is denoted by $n \times P$. It means

$$n \times P = \underbrace{P + P + \cdots + P}_{n}.$$

The result is also a point over G.

Obviously, computing a scalar multiplication with a large n is much more time-consuming than computing a point addition between two points.

3) Difficult Problems: Let P be the generator of G, and X, Y and W be points on G such that $X = \mathbf{x}P$, $Y = \mathbf{y}P$ and $W = \mathbf{w}P$ for some unknown $\mathbf{x}, \mathbf{y}, \mathbf{w} \in [0, n-1]$. Problems over elliptic curve groups are defined as follows.

- Computational Diffie-Hellman (CDH) Problem. Given (P, X, Y), the CDH problem over G is to find the point $W = \mathbf{xy}P$.
- Decisional Diffie-Hellman (DDH) Problem. Given (P, X, Y, W), the DDH problem over G is to determine whether or not $W = \mathbf{xy}P$.
- Gap Diffie-Hellman (GDH) Problem. Given (P, X, Y) and an oracle that solves the DDH problem over G, the GDH problem over G is to compute $W = \mathbf{xy}P$.

The hardness of CDH and GDH problems over G are assumed in this paper. It underlies the security of the proposed protocols.

B. ID-based System

1) ID-based Cryptography: ID-based cryptography embeds the user's identity in the public key to remove the use of public key certificates. The identity can be the name, identification number or email address of the user. In addition to users, an ID-based system involves a trusted authority called Key Generate Center (KGC). KGC establishes system parameters and extracts ID-based private keys for users. ID-based private keys are sent to users through secure channels.

2) ID-AKE: An ID-AKE protocol is composed of the following three procedures:

- Initialization. This procedure is executed by KGC and users. It normally involves the following phases:
  - Firstly, KGC inputs a security parameter and outputs the master key and system parameters. The master key is secretly kept by KGC. The system parameters are published to all the users. This phase is denoted as "KGC Setup".
  - Secondly, given the identity of a user, KGC extracts the ID-based private key for the user. The private key is securely sent to the user. This phase is denoted as "Key Extract".
  - Thirdly, the user stores the system parameters and secretly keeps the private key.
- Key Agreement. This procedure is executed by users (which are often called communicating parties). These parties generate short-term keys, exchange values over public channels and compute the shared secret respectively.
- Session Key Derivation. This procedure is executed by the parties. The parties derive session keys from the shared secret key.

The initialization procedure is not executed in every run of the protocol. The key agreement procedure is the core of an ID-AKE protocol.

C. Security Proof in mBR Model

Security of the proposed protocols are proved under random oracle model, in particular, the modified Bellare-Rogaway

TABLE I
COMPARISON OF AKE PROTOCOLS IN RELATED WORK.

Name of Scheme	Advantages	Disadvantages
Symmetric-AKE Protocols	Lightweight	(1) Requiring pre-shared secrets; (2) Equal computations on the two parties
Asymmetric-AKE Protocols	Not requiring pre-shared secret	(1) Requiring PKI; (2) Equal computations on the two parties
Pairing-based ID-AKE Protocols	Not requiring pre-shared secret or PKI	(1) Involving time-consuming pairing operations; (2) Equal computations on the two parties
Pairing-free ID-AKE Protocols	Not requiring pre-shared secret or PKI	Equal computations on the two parties

(mBR) model [25] via Computational No Reveal-mBR (cNR-mBR) game [27].

1) mBR Model: The mBR model is a well-defined model for the security of AKE protocols. It models a protocol as a pair $\mathcal{P} = (\Pi, \mathcal{G})$ where Π specifies behavior of honest parties, and $\mathcal{G}$ generates key pairs. Let 1^k be the security parameter, i and j be the identities of sender and receiver respectively, K_{ij} be i's key pair together with j's public key, $tran$ be the transcript of the protocol run so far. The notation $\Pi(1^k, i, j, K_{ij}, tran) = (m, \delta, K)$ denotes the execution of the protocol, where m is the next message from i to j; $\delta \in \{\text{Accept}, \text{Reject}, *\}$ is i's current decision; and K is the agreed session key.

2) cNR-mBR Game: The cNR-mBR game is a modular approach to prove security of AKE protocols in mBR model. The abilities and behaviors of an adversary $\mathcal{E}$ is specified as follows:

- $\mathcal{E}$ has access to a collection of oracles Π_{ij}^s which means the sth protocol running between i and j.
- $\mathcal{E}$ is allowed to make a polynomial number of $Send$ and $Corrupt$ queries to any oracle in any order.
- At the end of the game, $\mathcal{E}$ must choose an accepted and fresh oracle Π_{ij}^s [1]; and to win the game, $\mathcal{E}$ must compute the shared key.

$Advantage^{\mathcal{E}}(k)$ is defined as the probability that $\mathcal{E}$ outputs a session key equals to the agreed session key of Π_{ij}^s.

3) cNR-mBR-Secure: Based on the cNR-mBR game, the protocol security in mBR model is defined as follows.

Definition 2 (cNR-mBR-Secure): A protocol Π is cNR-mBR-secure if:

- In the presence of a benign adversary, two oracles running the protocol both accept holding the same session ID and session key which is distributed uniformly on $\{0,1\}^k$; and
- For any adversary $\mathcal{E}$, $Advantage^{\mathcal{E}}(k)$ in the cNR-mBR game is negligible.

4) Procedures of Proof: The security proof in mBR model via cNR-mBR game is conducted as follows:

- First, transform the target protocol Π into a related protocol Π' which is identical to Π, except that the session key produced by Π is the hashed output of the session key produced by Π'.
- Second, prove the security of Π' according to Definition 2.
- Finally, deduce the security of Π from the security of Π' according to the following Theorem 1.

Theorem 1: Let Π be an AKE protocol producing a hashed session key (via hash function H); and suppose that Π has strong partnering (Definition 3) and H is a random oracle. If the cNR-mBR security of the related protocol Π' is probabilistic polynomial time reducible to the hardness of the computational problem of some relation f; and the session string decisional problem for Π is polynomial time reducible to the decisional problem of f; then the mBR security of Π is probabilistic polynomial time reducible to the hardness of the Gap problem of f.

Definition 3 (Strong Partnering): A protocol has weak partnering if there exists an adversary $\mathcal{E}$ to the protocol such that $Advantage^{\mathcal{E}}(k)$ is non-negligible; and $\mathcal{E}$ can make any two oracles accept holding the same session key when they are not partners. A protocol has strong partnering if it does not have weak partnering.

D. Casper/FDR

The proposed protocols are also formally verified using Casper/FDR tool [28] which is a model checker for the process algebra Communicating Sequential Processes (CSP) [29]. Casper/FDR has been used to find previously unknown attacks on some famous security protocols such as the Needham-Schroeder public key protocol [30].

The Casper/FDR tool can verify secrecy and authenticity of security protocols under various attack model; and the attack model of our protocols is Dolev-Yao model [31].

Verifying protocols via Casper/FDR includes three procedures. First, rewrite the protocol and use Casper standard input script file to describe it. Second, compile the script file in Casper and output the CSP code file. Finally, check the CSP code via FDR.

IV. THE UNBALANCED AND PAIRING-FREE ID-AKA PROTOCOLS

In this section we first present an overview for the proposed ID-AKE protocols. Secondly, we present two unbalanced and pairing-free ID-AKE protocols which unbalance the computational task of the protocol in [14]. Finally, we compare the proposed protocols with existing protocols in related work.

A. Overview

1) Network Model: The ID-AKE protocol involves two parties that are connected by wireless channels. We specify

[1] An oracle is unfresh if it is revealed, or it has a revealed partner, or if its partner was corrupted. If an oracle is not unfresh, then the oracle is fresh.

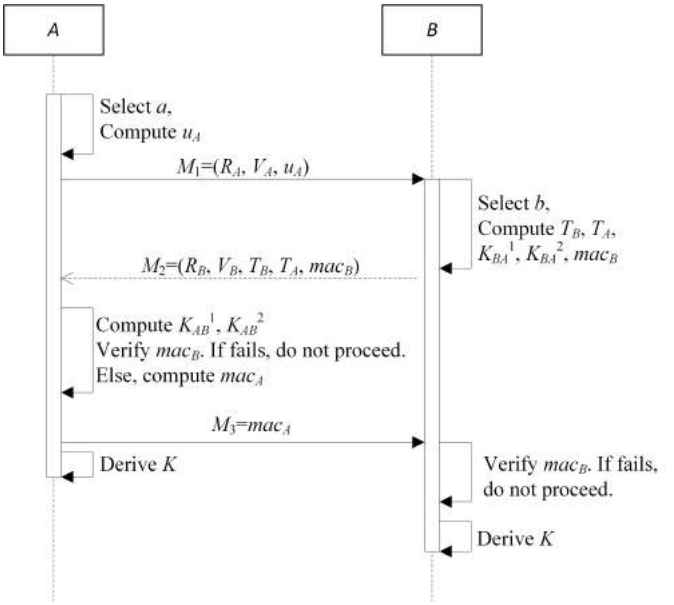

Fig. 1. Protocol I.

A as the initiating party, and B as the responding party. A and B have different computational capabilities.

2) Threat Model: The wireless channels between A and B are normal Dolev-Yao channels where the messages can be overheard, deleted or modified by the attacker. Specifically, the attackers have the following abilities.

- Basic ability. The attackers are able to observe, delete, insert, delay or alter messages between A and B.
- Stronger ability 1. The attackers are able to obtain any previous session keys.
- Stronger ability 2. The attackers are able to compromise the long-term secret keys of A or B.

B. Protocol I

Protocol I reduces computational cost on the initiator A by transferring one scalar multiplication from A to the responder B. Therefore, it has better performance when the responder is more powerful than the initiator. The protocol is presented as follows. It is also illustrated in Figure 1.

1) Initialization :

- KGC Setup: Given a security parameter k, KGC initializes system parameters $params = \{F_p, E, G, P, P_{pub}, H_1, H_2\}$ and the master-key $\mathbf{x}$ as follows.
  - F_p, E and G are the same as we defined in Definition 1.
  - P is a generator of G.
  - x is a random value over Z_p^* and $P_{pub} = x \times P$.
  - H_1 and H_2 are two cryptographic secure hash functions. H_1 maps an arbitrary string to Z_p^*. H_2 maps an arbitrary string to $\{0,1\}^k$.

  The system parameters are published to all the users and the master-key is kept secretly by KGC.
- Key Extract: Given the identity $ID \in \{0,1\}^*$, KGC computes the corresponding ID-based private key as follows.
  - Generate a random value $r_{ID} \in Z_p^*$ and compute
  $$R_{ID} = r_{ID} \times P,$$
  $$h_{ID} = H_1(ID \| R_{ID})$$
  and
  $$s_{ID} = r_{ID} + h_{ID} x.$$
  - Send (s_{ID}, R_{ID}) to the user through a secure out-of-band channel.
- User Setup: Upon receiving (s_{ID}, R_{ID}), the user firstly validates it through the following equation.
  $$s_{ID} \times P = R_{ID} + H_1(ID \| R_{ID}) \times P_{pub}.$$
  If the equation holds, (s_{ID}, R_{ID}) are valid. Then the user chooses a random value $v_{ID} \in Z_p^*$ and computes
  $$V_{ID} = v_{ID} \times P.$$

After the initialization, both A and B have the public information $params = \{F_p, E, G, P, P_{pub}, H_1, H_2\}$. Besides, A and B respectively hold (s_A, R_A, v_A, V_A) and (s_B, R_B, v_B, V_B). The values of s_A, s_B, v_A and v_B should be securely stored. Both A and B know the identity of each other before the key agreement procedure.

2) Key Agreement:

- A firstly generates a random value $a \in Z_p^*$ and computes $u_A = a + v_A$. Then A sends the following message M_1 to B.
  $$A \to B : M_1 = (R_A, V_A, u_A).$$
- Upon receiving M_1, B firstly generates a random value $b \in Z_p^*$ and computes $u_B = b + v_B$ and $T_B = u_B \times P$. Secondly, B computes the shared secrets as follows.
  $$T_A = u_A \times P$$
  $$K_{BA}^1 = s_B \times (T_A - V_A) + b \times (R_A + H_1(A \| R_A) \times P_{pub}),$$
  $$K_{BA}^2 = b \times (T_A - V_A).$$
  Thirdly, B computes mac_B and sends M_2 to A.
  $$mac_B = HMAC(K_{BA}^1 \| K_{BA}^2, R_B \| V_B \| T_B \| T_A)$$
  where $\|$ denotes the concatenation of bit strings.
  $$B \to A : M_2 = (R_B, V_B, T_B, T_A, mac_B).$$
- Upon receiving M_2, A firstly computes the shared secrets as follows.
  $$K_{AB}^1 = s_A \times (T_B - V_B) + a \times (R_B + H_1(B \| R_B) \times P_{pub}),$$
  $$K_{AB}^2 = a \times (T_B - V_B).$$
  Secondly, A verifies mac_B as follows.
  $$\text{VER}(K_{AB}^1 \| K_{AB}^2, R_B \| V_B \| T_B, mac_B) = \begin{cases} 1, & valid \\ 0, & invalid \end{cases}$$
  Thirdly, if mac_B is valid, A computes mac_A and sends M_3 to B.
  $$mac_A = HMAC(K_{AB}^1 \| K_{AB}^2, R_A \| V_A \| u_A)$$

$$A \rightarrow B : M_3 = mac_A$$

- Upon receiving M_3, B verifies mac_A as follows.

$$\text{VER}(K_{BA}^1 \| K_{BA}^2, R_A \| V_A \| u_A, mac_A) = \begin{cases} 1, & valid \\ 0, & invalid \end{cases}$$

$K_{AB}^1 = K_{BA}^1$ and $K_{AB}^2 = K_{BA}^2$ because:

$$\begin{aligned} K_{AB}^1 &= s_A \times (T_B - V_B) + a \times (R_B + H_1(B \| R_B) \times P_{pub}) \\ &= s_A \times (u_B \times P - V_B) + as_B \times P \\ &= s_A \times (b \times P + v_B \times P - V_B) + as_B \times P \\ &= s_A b \times P + s_B a \times P \end{aligned}$$

$$\begin{aligned} K_{BA}^1 &= s_B \times (T_A - V_A) + b \times (R_A + H_1(A \| R_A) \times P_{pub}) \\ &= s_B \times (u_A \times P - V_A) + bs_A \times P \\ &= s_B \times (a \times P + v_A \times P - V_A) + bs_A \times P \\ &= s_B a \times P + s_A b \times P \\ &= K_{AB}^1 \end{aligned}$$

$$\begin{aligned} K_{AB}^2 &= a \times (T_B - V_B) \\ &= a \times (u_B \times P - V_B) \\ &= a \times (b \times P + v_B \times P - V_B) \\ &= ab \times P \end{aligned}$$

$$\begin{aligned} K_{BA}^2 &= b \times (T_A - V_A) \\ &= b \times (u_A \times P - V_A) \\ &= b \times (a \times P + v_A \times P - V_A) \\ &= ba \times P \\ &= K_{AB}^2 \end{aligned}$$

3) Session Key Derivation: If mac_A and mac_B are valid, A and B compute the session key as follows.

A computes

$$sk_{AB} = H_2(A \| B \| T_A \| T_B \| K_{AB}^1 \| K_{AB}^2).$$

B computes

$$sk_{BA} = H_2(A \| B \| T_A \| T_B \| K_{BA}^1 \| K_{BA}^2).$$

C. Protocol II

Protocol II reduces computational cost on B by transferring one scalar multiplication from B to A. The protocol is presented as follows. Therefore, it has better performance when the initiator is more powerful than the responder. The protocol is presented as follows. It is also illustrated in Figure 2.

1) Initialization : The initialization procedure is exactly the same with that in Section IV-B.

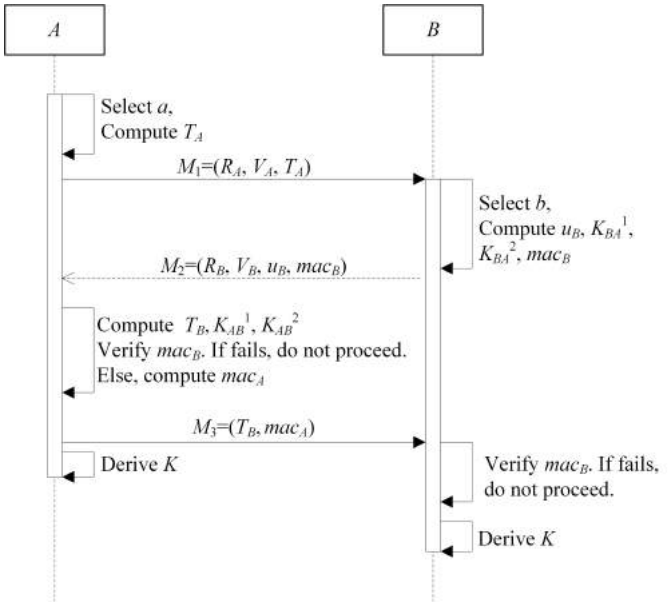

Fig. 2. Protocol II.

2) Key Agreement:

- A firstly generates a random value $a \in Z_p^*$ and computes $u_A = a + v_A$ and $T_A = u_A \times P$. Then A sends the following message M_1 to B.

$$A \rightarrow B : M_1 = (R_A, V_A, T_A).$$

- Upon receiving M_1, B firstly generates a random value $b \in Z_p^*$ and computes $u_B = b + v_B$.
Secondly, B computes the shared secrets as follows.

$$K_{BA}^1 = s_B \times (T_A - V_A) + b \times (R_A + H_1(A \| R_A) \times P_{pub}),$$

$$K_{BA}^2 = b \times (T_A - V_A).$$

Thirdly, B computes mac_B and sends M_2 to A.

$$mac_B = HMAC(K_{BA}^1 \| K_{BA}^2, R_B \| V_B \| u_B)$$

$$B \rightarrow A : M_2 = (R_B, V_B, u_B, mac_B).$$

- Upon receiving M_2, A firstly computes the shared secrets as follows.

$$T_B = u_B \times P,$$

$$K_{AB}^1 = S_A \times (T_B - V_B) + a \times (R_B + H_1(B \| R_B) \times P_{pub}),$$

$$K_{AB}^2 = a \times (T_B - V_B).$$

Secondly, A verifies mac_B as follows.

$$\text{VER}(K_{AB}^1 \| K_{AB}^2, R_B \| V_B \| u_B, mac_B) = \begin{cases} 1, & valid \\ 0, & invalid \end{cases}$$

Thirdly, if mac_B is valid, A computes mac_A and sends M_3 to B.

$$mac_A = HMAC(K_{AB}^1 \| K_{AB}^2, R_A \| V_A \| T_A \| T_B)$$

$$A \rightarrow B : M_3 = (T_B, mac_A)$$

- Upon receiving M_3, B verifies mac_A as follows.

$$\text{VER}(K_{BA}^1 \| K_{BA}^2, R_A \| V_A \| T_A \| T_B, mac_A) = \begin{cases} 1, & valid \\ 0, & invalid \end{cases}$$

3) Session Key Derivation: If mac_A and mac_B are valid, A and B compute the session key as follows. A computes

$$sk_{AB} = H_2(A \| B \| T_A \| T_B \| K_{AB}^1 \| K_{AB}^2).$$

B computes

$$sk_{BA} = H_2(A \| B \| T_A \| T_B \| K_{BA}^1 \| K_{BA}^2).$$

D. Comparison with Existing Protocols

We compare the proposed protocols with existing protocols in Table III and IV. Detailed analysis and evaluation will be provided in the following three sections.

Table III compares the security of the proposed protocols with typical ID-AKE protocols in related work. According to the table, the proposed protocols have provable security in mBR model; and their security have been formally verified by the Casper/FDR tool. Therefore, in terms of security, they are better than most of the other protocols compared in the table.

In Table IV, different types of AKE protocol are compared in terms of functionality features. According to the table, the proposed protocols do not rely on any pre-shared secret or PKI, and have unbalanced cost on the parities. This makes them have better performance in situations where the communicating parties have different computational capabilities.

V. Security Analysis

Security of Protocol I is studied through three methods. We first prove the security of Protocol I under random oracle model through mBR model. Secondly, we formally verify the protocol using Casper/FDR. Finally, we analyze its resistance to some attacks. Security of Protocol II is similar with that of Protocol I; therefore, we do not provide detailed analysis in this paper.

A. Security Proof

Security of Protocol I in mBR model is proved through the following three procedures.

1) Transform Protocol I to I': Protocol I' is identical to Protocol I except the session key derivation procedure. Protocol I' uses the string $(A \| B \| T_A \| T_B \| K_{AB}^1 \| K_{AB}^2)$ as the session key.

2) Prove Security of Protocol I': The cNR-mBR security of Protocol I' is proved as follows.

Theorem 2: Assume that H_1 is a random oracle. Given the security parameter k of Protocol I', if there is an adversary $\mathcal{E}$ to Protocol I' can win the cNR-mBR game with non-negligible probability σ in polynomial time τ, then the CDH problem can be solved with non-negligible probability $\eta \frac{1}{n_p^2 n_s} \sigma$ within time τ where η is a constant.

Proof: Suppose $\mathcal{A}$ is an algorithm to solve the CDH problem. $\mathcal{A}$ is given an instance (X, Y) for some unknown $\mathbf{x}, \mathbf{y} \in [0, n-1]$, and is asked to compute W such that $W = \mathbf{xy}P$. To do this, $\mathcal{A}$ simulates a challenger in a cNR-mBR game with $\mathcal{E}$. $\mathcal{A}$ stipulates the hash function H_1 and maintains an H_1-list which is initialized empty. Let n_p denotes the number of participants in the game, and n_s denotes the number of sessions each participant may be involved in. The long-term key for the ith participant ID_i is (s_i, R_i, v_i, V_i) and ID_i is the corresponding public key. $\mathcal{A}$ randomly chooses $P_0 \in G$, sets P_0 as P_{pub}, and generates ID_i's long-term key as follows:

- First, $\mathcal{A}$ randomly chooses I from $[1, n_p]$ and generates the long-term key for ID_I. In particular, $\mathcal{A}$ randomly chooses $h_I, v_I \in Z_p^*$ and $V_I \in G$, computes $R_I = Y - h_I P_0$, and sets $(\perp, R_I, v_I, V_I)$ as the long-term key.
- Second, $\mathcal{A}$ generates long-term key for ID_i where $i \in [1, n_p]$ and $i \neq I$. In particular, $\mathcal{A}$ randomly chooses $s_i, h_i, v_i \in Z_p^*$ and $V_i \in G$, computes $R_i = s_i P - h_I P_0$, and sets (s_i, R_i, v_i, V_i) as the long-term key.
- Third, $\mathcal{A}$ passes R_i and ID_i to $\mathcal{E}$ and adds $\{ID_i, R_i, h_i\}$ to the H_1-list for $i = 1, ..., n_p$.

After generating the long-term keys, $\mathcal{A}$ randomly chooses $J \in [1, n_p] \neq I$ and $\mathbf{v} \in [1, n_s]$, and starts $\mathcal{E}$ by answering $\mathcal{E}$'s queries as follows.

- $H_1(ID_i, R_i)$: If $\{ID_i, R_i, h_i\}$ is in the H_1-list, $\mathcal{A}$ responses with h_i; otherwise, $\mathcal{A}$ randomly chooses $l_i \in Z_p^*$, responses with l_i, and adds $\{ID_i, R_i, l_i\}$ to the H_1-list.
- $Send(\mathrm{I}_{i,j}^s, M)$: If $\mathrm{I}_{i,j}^s \neq \mathrm{I}_{J,j}^{\mathbf{v}}$, $\mathcal{A}$ acts according to the protocol specification; otherwise, $\mathcal{A}$ responds with $(ID_j, R_j, X + V_j)$.
- $Corrupt(i)$: If $i = I$, then $\mathcal{A}$ aborts; otherwise, $\mathcal{A}$ returns s_i to $\mathcal{E}$.

The probability that $\mathcal{E}$ chooses $\mathrm{I}_{J,j}^{\mathbf{v}}$ as the Test oracle and that $ID_j = ID_I$ is $\frac{1}{n_p^2 n_s}$. In this case, $\mathcal{E}$ would not have corrupted ID_I; therefore $\mathcal{A}$ would not have aborted. If $\mathcal{E}$ can win in such a $cNR - mBR$ game with probability σ, then at the end of the game, $\mathcal{E}$ will output its guess of the session key in the form $(\{0,1\}^* \| \{0,1\}^* \| X + V_j \| Y \| Z \| W)$; and $\mathcal{A}$ can output W as its solution to the CDH problem on input (X, Y) with non-negligible probability $\eta \frac{1}{n_p^2 n_s} \sigma$ within time τ where η is a constant. ■

3) Prove Security of Protocol I: We first prove that Protocol I satisfies the property of strong partnering.

Theorem 3: Protocol I has strong partnering in the random oracle model.

Proof: Denote the partner of user i by i'. Assume there exists an adversary $\mathcal{E}$ who can make two oracles $\mathrm{I}_{i,i'}^s$ and $\mathrm{I}_{j,j'}^t$ accept holding the same session key when $i' \neq j$ and $j' \neq i$. If $\mathrm{I}_{i,i'}^s$ is an initiator, to obtain the session key, it has to make a query of the form $\{ID_i, ID_{i'}, X, Y, Z, W\}$ to the H_2 random oracle. For $\mathrm{I}_{j,j'}^t$ to have the same session key, it must have made the H_2 query of the form $\{ID_j, ID_{j'}, X, Y, Z, W\}$ since $ID_j \neq ID_i$. Then it must have $ID_{j'} = ID_i$ and vice versa. Thus $\mathrm{I}_{i,i'}^s$ and $\mathrm{I}_{j,j'}^t$ are partners, which contradicts the assumption.

Thus it is impossible for $\mathcal{E}$ to obtain a qualified $\mathrm{I}_{j,j'}^t$ when $\mathrm{I}_{i,i'}^s$ is an initiator. Similar we can prove this is impossible

TABLE III
COMPARISON OF THE PROPOSED PROTOCOLS WITH TYPICAL ID-AKE PROTOCOLS IN RELATED WORK.

AKE Protocol	Category	Provable Security	Formal Verification
Protocol in [10]	Pairing-based	×	×
Protocol in [11]	Pairing-based	×	×
Protocol in [12]	Pairing-based	in BR model	×
Protocol in [13]	Unbalanced pairing-based	×	×
Protocol in [14]	Pairing-free	in mBR model	×
Protocol in [15]	Pairing-free	in eCK model	×
Protocol in [16]	Pairing-free	in eCK model	×
Protocol I	Unbalanced and pairing-free	in mBR model	using Casper/FDR tool
Protocol II	Unbalanced and pairing-free	in mBR model	using Casper/FDR tool

TABLE IV
COMPARISON OF DIFFERENT TYPES OF AKE PROTOCOLS.

Category of AKE	Requiring Pre-shared Secret	Requiring PKI	Having Unbalanced Cost
Symmetric-AKE [17][18][19]	√	×	×
Asymmetric-AKE [20][21]	×	√	×
Pairing-based ID-AKE [10][11][12]	×	×	×
Unbalanced and pairing-based ID-AKE [13]	×	×	√
Pairing-free ID-AKE [14][15][16]	×	×	×
Unbalanced and pairing-free ID-AKE	×	×	√

when $\mathrm{I}^s_{i,i'}$ is a responder. Therefore, Protocol I has strong partnering. ■

According to Theorem 1, 2 and 3, we have the following conclusion: Protocol I is secure in the mBR model assuming that H_1 is a random oracle and the CDH and GDH problems are difficult.

B. Formal Verification

We use the Casper/FDR tool to verify Protocol I, in particular, the authentication of exchange messages in Protocol I.

Firstly, we rewrite Protocol I according to syntax of Casper standard input script as follows:

- $A \rightarrow B : ra, va, ua.$
- $B \rightarrow A : rb, vb, tb, ta, hash(kab, rb, vb, tb, ta).$
- $A \rightarrow B : hash(kab, ra, va, ua)$

In the above description, ra, va, ua, rb, vb, tb, ta and $hash$ correspond to R_A, V_A, u_A, R_B, V_B, T_B, T_A and $HMAC$, and kab corresponds to $K^1_{AB} \| K^2_{AB}$ in Protocol I.

Secondly, we use Casper to compile the script into CSP code. The CSP code which can be run directly in FDR is uploaded to GitHub[2]. Finally, we verify the CSP code via FDR. The result is shown in Figure 3. According to the figure, no attack is found. Therefore, the authentication of R_A, V_A, u_A, R_B, V_B and T_B are guaranteed in Protocol I. This means attackers are unable to launch the man-in-the-middle attack to the protocol.

[2]https://github.com/cuttercn/upfidakeProtocol

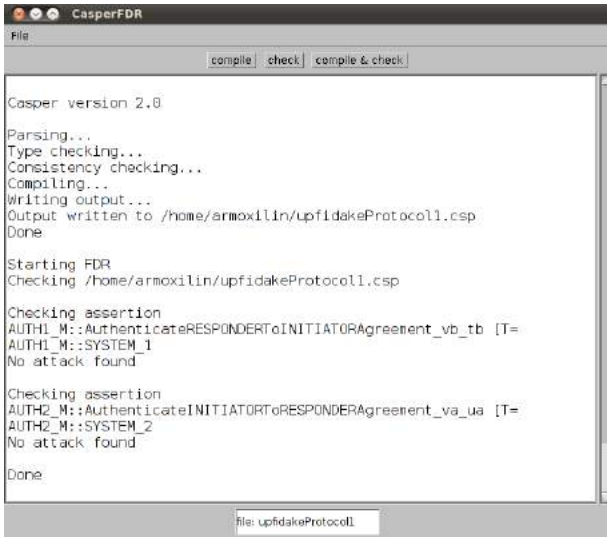

Fig. 3. FDR checking result of Protocol I.

C. Resistance to Attacks

We analyze how Protocol I is resistant to some commonly encountered attacks as follows.

1) Resistance to Impersonation Attacks: Assume $\mathcal{E}$ is an impersonation attacker to Protocol I. The long-term key of $\mathcal{E}$ is $(s_\mathcal{E}, R_\mathcal{E}, v_\mathcal{E}, V_\mathcal{E})$. To launch the attack, $\mathcal{E}$ impersonates A and interacts with B as follows.

- $\mathcal{E}$ firstly generates a random value $e \in Z_p*$ and computes $u_\mathcal{E} = e + v_\mathcal{E}$. Then $\mathcal{E}$ sends the following message M_1 to B.

$$\mathcal{E} \rightarrow B : M_1 = (R_\mathcal{E}, V_\mathcal{E}, u_\mathcal{E}).$$

- Upon receiving M_1, B firstly generates a random value $b \in Z_p^*$ and computes $u_B = b + v_B$ and $T_B = u_B \times P$. Secondly, B computes the shared secrets as follows.

$$T_{\mathcal{E}} = u_{\mathcal{E}} \times P$$

$$K^1_{B\mathcal{E}} = s_B \times (T_{\mathcal{E}} - V_{\mathcal{E}}) + b \times (R_{\mathcal{E}} + H_1(A\|R_{\mathcal{E}}) \times P_{pub}),$$

$$K^2_{B\mathcal{E}} = b \times (T_{\mathcal{E}} - V_{\mathcal{E}}).$$

Thirdly, B computes mac_B and sends M_2 to $\mathcal{E}$.

$$mac_B = HMAC(K^1_{B\mathcal{E}}\|K^2_{B\mathcal{E}}, R_B\|V_B\|T_B\|T_{\mathcal{E}})$$

$$B \rightarrow A : M_2 = (R_B, V_B, T_B, T_{\mathcal{E}}, mac_B).$$

- Upon receiving M_2, $\mathcal{E}$ firstly computes the shared secrets as follows.

$$K^1_{\mathcal{E}B} = s_{\mathcal{E}} \times (T_B - V_B) + e \times (R_B + H_1(B\|R_B) \times P_{pub}),$$

$$K^2_{\mathcal{E}B} = e \times (T_B - V_B).$$

Secondly, $\mathcal{E}$ needs to compute a $mac_{\mathcal{E}}$ which can pass the verification of B in the following step. $\mathcal{E}$ is unable to compute such a $mac_{\mathcal{E}}$ from $K^1_{\mathcal{E}B}$ and $K^2_{\mathcal{E}B}$ since $K^1_{\mathcal{E}B} \neq K^1_{B\mathcal{E}}$.
- In this step, the verification of $mac_{\mathcal{E}}$ fails; and B terminates the protocol. The attack fails.

Similarly, if $\mathcal{E}$ impersonates B, the attack will fail.

2) Forward Secrecy under Stolen Device Attacks: Forward secrecy guarantees that the attacker who corrupts the long-term keys of the participants cannot reveal session keys in previous sessions. It is a significant security feature, in particular, when the device is stolen by the attacker. In this case, the participants can reset the long-term keys; and previous communicating messages will be secure if forward secrecy is provided. Bellow we will analyze how Protocol I provide forwards secrecy under the stolen device attack.

In Protocol I, the session key sk_{AB} of a previous session is derived from K^1_{AB} and K^2_{AB}; and the computation of K^1_{AB} and K^2_{AB} involve two random values a and b. The random values are generated in every session of the protocol, and are disposed after the session. As a result, $\mathcal{E}$ is unable to acquire a and b of a pervious session. Therefore, $\mathcal{E}$ cannot reveal previous session keys.

VI. Performance

In order to study the performance of the proposed protocols, we first evaluate their computational costs and compare with existing pairing-free ID-AKE protocols. Then, two sets of experiments are carried out. Experiment I is used to study the computational cost. It verifies whether Protocol I and II have unbalanced computational requirements and are more friendly to limited devices than existing pairing-free ID-AKE protocols. Experiment II is used to study the availability of the protocols in disaster scenarios. It verifies whether Protocol I and II are more convenient and suitable for disaster scenarios than symmetric and traditional asymmetric cryptography based AKE protocols.

TABLE V
Cost evaluation.

Protocol	Scalar multiplication on A	Scalar multiplication on B
[14]	5	5
[15]	6	6
[16]	6	6
Protocol I	4	6
Protocol II	6	4

TABLE VI
Experimental environment of Experiment I-1.

Party	Operating System	Base Memory	Storage
A	Ubuntu 16.04.3 (32-bit)	1024 MB	10 GB
B	Ubuntu 16.04.3 (32-bit)	1024 MB	10 GB

A. Evaluation

The computational cost is evaluated by the number of scalar multiplication on each party (Table V).

According to Table V, Protocol I has the lowest computational cost on A; and Protocol II has the lowest computational cost on B. Therefore, when A is a limited device, Protocol I is the most suitable ID-AKE protocol; when B is a limited device, Protocol II is the most suitable ID-AKE protocol.

B. Experiment I: Computational Cost

We realize prototypes of Protocol I, II and typical pairing-free ID-AKE protocols in [14] and [15] using Python programming language. Two experiments are carried out; and computational time is tested to observe the computational cost. The experiments are introduced as follows.

1) Experiment I-1: Experiment I-1 is used to verify whether Protocol I and II have unbalance computational requirements. In the experiment, we deploy the initiator A and the responder B on two virtual machines with the same configuration (Table VI). We have run the prototype of each protocol for ten times on three of recommended elliptic curves in Federal Information Processing Standards (FIPS), i.e., P-192, P-256 and P-384. The average computational time is illustrated in Figure 4.

According to Figure 4, for all of the three curves, Protocol I has the lowest computational time on A. Protocol II has the lowest computational time on B. We have the following conclusions:

- Protocol I has lower computational requirement on A than on B.
- Protocol II has lower computational requirement on B than on A.

2) Experiment I-2: Experiment I-2 is used to verify whether Protocol I and II are more friendly to limited devices than the protocols in [14] and [15]. In this experiment, we use a Raspberry Pi to simulate the limited device and a laptop to simulate the powerful one. For Protocol I, the initiator A is deployed on the Raspberry Pi. For Protocol II, the

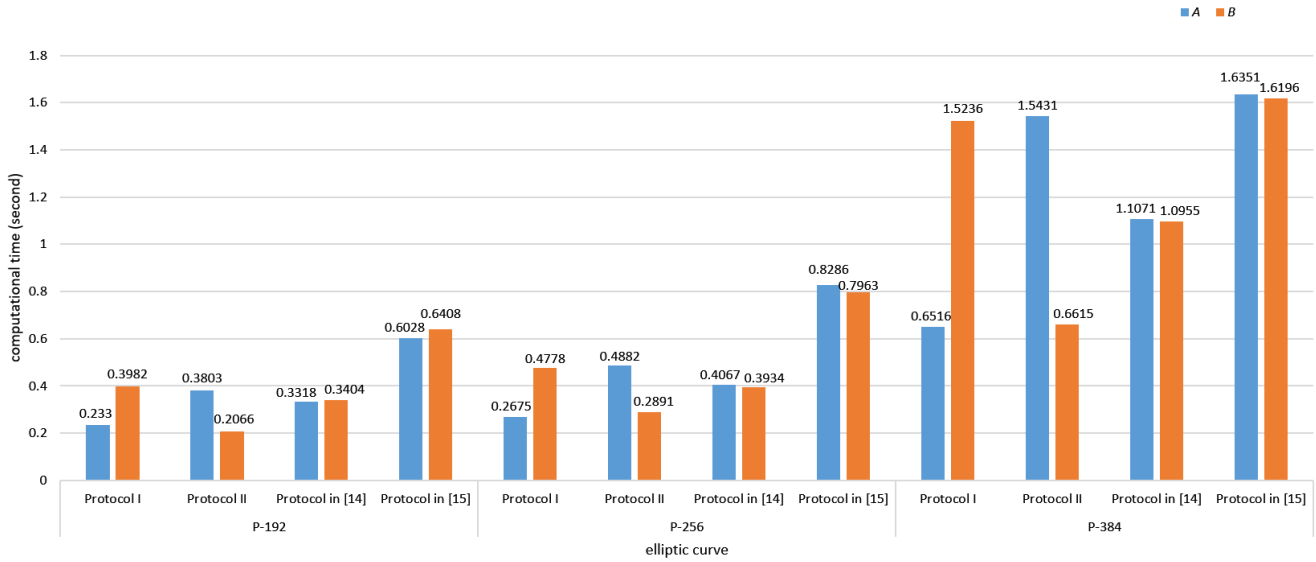

Fig. 4. Average computational time in Experiment I-1.

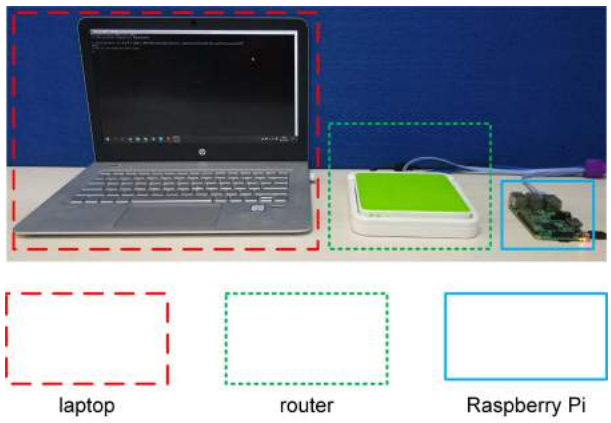

Fig. 5. Hardware platform of Experiment I-2.

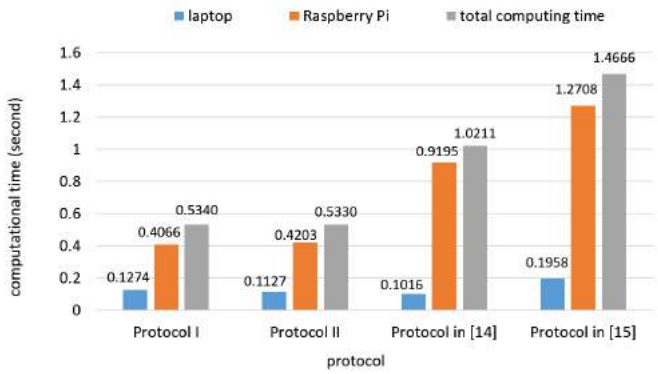

Fig. 6. Average computational time in Experiment I-2.

responder B is deployed on the Raspberry Pi. The details of the experimental environment are listed in Table VII. The hardware platform is illustrated in Figure 5. We have run the prototypes for ten times on P-256. The average runtime is illustrated in Figure 6.

According to Figure 6, the computational time on the Raspberry Pi of Protocol I and II are much less than that of the protocols in [14] and [15]; the overall computational time of Protocol I and II are also much less than that of the protocols in [14] and [15]. We have the following two conclusions:

- Protocol I and II are more friendly to limited devices than the protocols in [14] and [15].
- In terms of computational time, the overall performance of Protocol I and II are better than that of the protocols in [14] and [15]

C. Experiment II: Availability

We evaluate the availability of Protocol I (or II), a symmetric cryptography-based AKE protocol in [17] (denoted as symmetric-AKE) and an asymmetric cryptography based AKE protocol in Transport Layer Security (TLS) standard [20] (denoted as TLS-AKE). The experiment is described as follows.

1) Assumption: Let N denote the set of devices (carried by $|N|$, i.e. the length of set the N , rescue officers) entering into a disaster area where the network infrastructures are destroyed; N_1 denote the subset of N that devices have shared master keys with each other; N_2 denote the subset of N that devices have public keys of each other; and N_3 denote the subset of N that devices neither have shared master keys nor have public keys of each other. When a device $A \in N_1$ (or $A \in N_2$) needs to establish a secure link with a device B, it at first tries the symmetric-AKE protocol (or TLS-AKE protocol). If $B \notin N_1$ (or $B \notin N_2$), they will then use Protocol I or II. The availability of symmetric-AKE protocol in [17], TLS-AKE protocol in [20] and the proposed Protocol I or II are evaluated by P_1, P_2 and P_3 as follows:

- $P_1 = Pr(A \in N_1 \cap B \in N_1) = (\frac{|N_1|}{|N|})^2$: the probability that two arbitrary devices A and B successfully execute the symmetric-AKE protocol.

TABLE VII
EXPERIMENTAL ENVIRONMENT OF EXPERIMENT I-2.

Experimental Device	CPU	Base Memory	Hard Disk
Raspberry Pi	1.2 GMHz ARM	1 GB	32 GB
laptop	2.40 GHz i5-6200U	4 GB	120 GB

- $P_2 = Pr(A \in N_2 \cap B \in N_2) = (\frac{|N_2|}{|N|})^2$: the probability that two arbitrary devices A and B successfully execute the TLS-AKE protocol.
- $P_3 = 1 - P_1 - P_2 = 1 - (\frac{|N_1|}{|N|})^2 - (\frac{|N_1|}{|N|})^2$: the probability that two arbitrary devices A and B successfully execute Protocol I or II.

2) Experiment: In the experiment, we generate 1000 groups of random value for $(|N_1|, |N_2|, |N_3|)$. The results of P_1, P_2 and P_3 are compared in Figure 7. According to Figure 7, for two arbitrary devices in the disaster area, in most cases, the success probability of Protocol I or II (P_3) is larger than that of symmetric-AKE protocol [17] and TLS-AKE protocol [20] (P_1 and P_2). Besides, when the proportion of N_3 in N increases, P_3 increases while P_1 and P_2 decrease.

Therefore, we have two conclusions as follows:

- Protocol I (or II) has a higher availability than symmetric-AKE protocol [17] and TLS-AKE protocol [20] in a disaster scenario.
- As the number of unacquainted devices (devices do not have shared master key or exchanged public key) increases, the availability of Protocol I (or II) grows while that of symmetric-AKE protocol [17] and TLS-AKE protocol [20] decrease.

It is inconvenient for devices (or rescuer officers) from different organizations to share master keys or have public keys with each other; and rescue officers for mass casualty disasters often come from various organizations (even different countries). Therefore, Protocol I and II show significant advantage in terms of availability in rescues for mass casualty disasters such as earthquake, typhoon and war.

VII. USE CASE

This section illustrates the application of Protocol I and II via a use case of rescuing in an affected area after a terrorist attack.

A. Scenario Description

We use a wasteland in the suburb to simulate the affected area after a terrorist attack. The power and network infrastructures are destroyed by the attack. A group of "rescuers" acted by our volunteers will enter the "affected area" (Figure 8).

At first, a police named Bob with his smart phone enters the "affected area". After ten minutes, an emergency medical technician named Alice with her laptop arrives at the "affected area". (In a real affected area, there will be mobile tents or ambulances for the emergency medical technician to place the laptop.) Meanwhile, a criminal named Malice with his tablet sneaks into the "affected area". This is quite possible in a real affected area since there can be several entrances to the area.

Now, all of Bob, Alice and Malice are in the "affected area". Protocol I and II will be used as the foundation to protect the communication between Alice's laptop and Bob's smart phone from being attacked by Malice's tablet.

B. Application of Protocol I and II

The following procedures illustrate how to apply Protocol I and II in protecting communications in the above scenario.

1) Setup: Both Alice's laptop and Bob's smart phone should be configured to support Protocol I and II. In particular, Alice's laptop is regarded as a powerful device; therefore, we configure it with the code of responder of Protocol I and that of initiator of Protocol II. Bob's smart phone is regarded as a limited device; therefore, we configure it with the code of initiator of Protocol I and that of responder of Protocol II.

In addition, Both Alice's laptop and Bob's smart phone also should be configured with encryption/decrytion schemes such as AES and message authentication code schemes such as HMAC.

2) Secure Handshake: Alice's laptop and Bob's smart phone collaboratively run Protocol I or II to establish a shared key. There are two cases:

- If Bob's smart phone initiates the communication with Alice's laptop, Bob's smart phone will run the code of initiator of Protocol I; and Alice's laptop will run the code of responder of Protocol I.
- If Alice's laptop initiates the communication with Bob's smart phone, Alice's laptop will run the code of initiator of Protocol II; and Bob's smart phone will run the code of responder of Protocol II.

At the end of secure handshake, a session key sk is established for Alice's laptop and Bob's smart phone.

3) Secure Messaging: When Alice's laptop intends to send a message M_{AB} to Bob's smart phone, her laptop first encrypts M_{AB} into $\{M_{AB}\}_{sk_{AB}}$ and computes a message authentication code mac_{AB} for $\{M_{AB}\}_{sk_{AB}}$. Then $\{M_{AB}\}_{sk_{AB}}$ is sent along with mac_{AB}. After receiving the message, Bob's smart phone first verifies mac_{AB}. If the verification succeeds, his smart phone decrypts $\{M_{AB}\}_{sk_{AB}}$ and acquires $\{M_{AB}\}$. When Bob's smart phone intends to send a message M_{AB} to Alice's laptop, the process is similar as above.

Without the session key sk_{AB}, Malice's tablet is unable to reveal, tamper or fake the message.

VIII. CONCLUSION

In this paper, we investigated suitable ID-AKE protocols in disaster scenarios. We proposed the idea to unbalance the computing tasks of existing pairing-free ID-AKE protocols. Based on the idea, we designed two unbalanced and pairing-free ID-AKE protocols that transfer one scalar multiplication

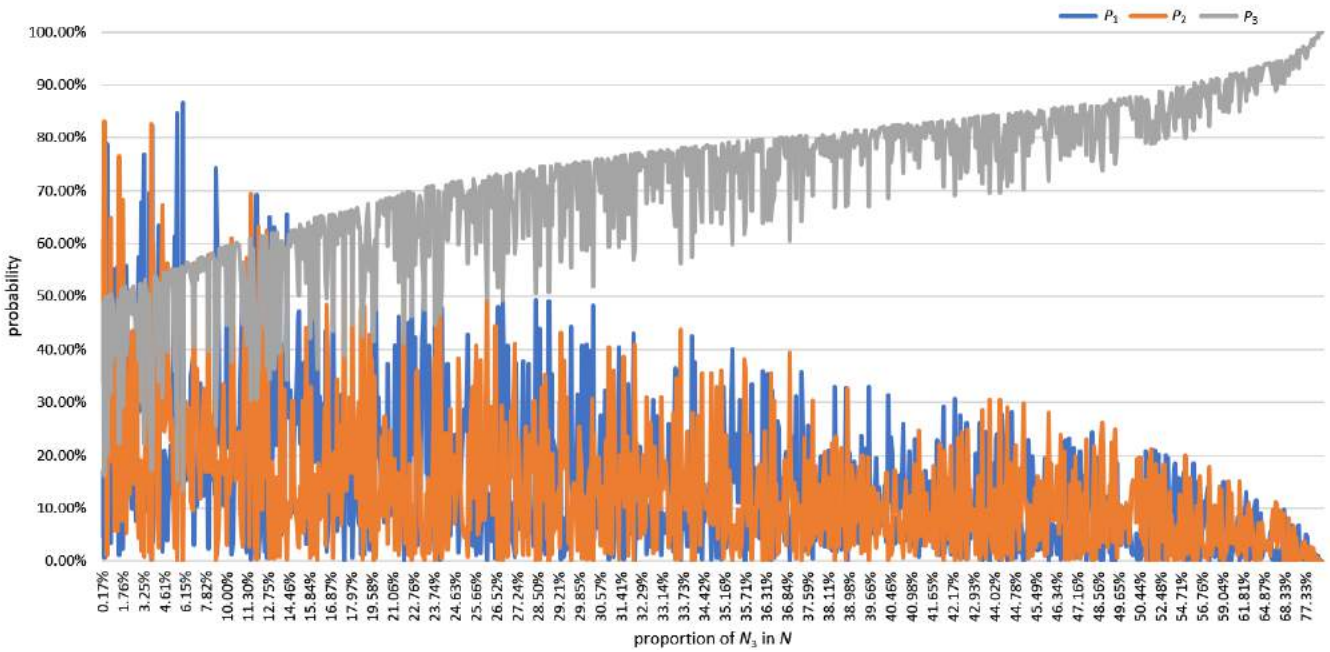

Fig. 7. Probabilities of success in terms of symmetric-AKE protocol [17], TLS-AKE protocol [20] and Protocol I or II in a disaster scenario.

from one side to another. Prototypes of the protocols were realized to study the performance. We also illustrated how to apply the proposed protocols via a use case.

The proposed idea can be used to unbalance the computational costs of other existing pairing-free ID-AKE protocols. In the future, we plan to modify some of existing pairing-free ID-AKE protocols into unbalanced ones. In addition, we are going to transfer more scalar multiplications from one side to another in pairing-free ID-AKE protocols.

ACKNOWLEDGMENT

The authors appreciate the help of Zilong Wei, Kai Zheng and Lei Chen to assist the use case part in this paper.

REFERENCES

[1] Intelligent Transportation Systems Committee, *512-2006 - IEEE standard for common incident management message sets for use by emergency management centers.* IEEE Standards, 2006.

[2] Intelligent Transportation Systems Committee, *1512.1-2003 - IEEE Standard for Traffic Incident Management Message Sets for Use by Emergency Management Centers.* IEEE Standards, 2003.

[3] Intelligent Transportation Systems Committee, *1512.2-2004 - IEEE Standard for Public Safety Traffic Incident Management Message Sets for Use By Emergency Management Centers.* IEEE Standards, 2004.

[4] Intelligent Transportation Systems Committee, *1512.3-2006 - IEEE Standard for Hazardous Material Incident Management Message Sets for Use by Emergency Management Centers.* IEEE Standards, 2006.

[5] A. R. McGee, M. Coutiere and M. E. Palamara, *Public Safety Network Security Considerations.* Bell Labs Technical Journal, 17(3), 79-86, 2012.

[6] D. J. Solove and P. Schwartz, *Information Privacy Law.* Wolters Kluwer Law & Business, 2014.

[7] A. M. Campillo, J. Crowcroft, E. Yoneki and R. Marti, *Evaluating Opportunistic Networks in Disaster Scenarios.* Journal of Network and Computer Applications, 36(2),870-880, 2013.

[8] N. Aschenbruck, M. Frank, P. Martini and J. Tolle, *Human Mobility in MANET Disaster Area Simulation - A Realistic Approach.* Local Computer Networks, 29th Annual IEEE International Conference on, 2004.

[9] M. Dou, J. Chen, D. Chen, X. Chen, Z. Deng, X. Zhang, K. Xu and J. Wang, *Modeling and Simulation for Natural Disaster Contingency Planning Driven by High-resolution Remote Sensing Images*. Future Generation Computer Systems, 37, 367-377, 2014.

[10] N. P. Smart, *Identity-based Authenticated Key Agreement Protocol Based on Weil Pairing.* Electronics Letters, 83(13), 630-632, 2002.

[11] K. Shim. *Efficient ID-based Authenticated Key Agreement Protocol Based on Weil Pairing.* Electronics Letters, 39(8), 653-654, 2003.

[12] L. Chen and C. Kudla, *Identity Based Authenticated Key Agreement Protocols from Pairings.* Computer Security Foundations Workshop, 2003. Proceedings. 16th IEEE, 2003.

[13] M. Scott, *Unbalancing Pairing-Based Key Exchange Protocols.* IACR Cryptology ePrint Archive, 688, 2013.

[14] X. Cao, W. Kou and X. Du, *A Pairing-free Identity-based Authenticated Key Agreement Protocol With Minimal Message Exchanges.* Information Sciences, 180, 2895-2903, 2010.

[15] L. Ni, G. Chen, J. Li and Y. Hao, *Strongly Secure Identity-based Authenticated Key Agreement Protocols Without Bilinear Pairings.* Information Sciences, 367, 176-193, 2016.

[16] L. Dang, J. Xu, X. Cao, H. Li, J. Chen, Y. Zhang and X. Fu, *Efficient Identity-based Authenticated Key Agreement Protocol with Provable Security for Vehicular Ad Hoc Networks.* International Journal of Distributed Sensor Networks, 14(4), 2018.

[17] M. Bellare and P. Rogaway, *Entity Authentication and Key Distribution.* Advances in Cryptology - Crypto '93, Lecture Notes in Computer Science, 773, 232-249, 1993.

[18] R. M. Needham and M. D. Schroeder, *Using Encryption for Authentication in Large Networks of Computers.* Communications of the ACM, 21(12), 993-999, 1978.

[19] B. C. Neuman and T. Ts'o, *Kerberos: An Authentication Service for Computer Networks.* IEEE Communications Magazine, 32(9), 33-38, 1994.

[20] T. Dierks and E. Rescorla, *The Transport Layer Security TLS Protocol Version 1.2.* The IETF Trust, RFC 5246, 2008.

[21] W. Diffie and M. E. Hellman, *New Directions in Cryptography.* IEEE Transactions on Information Theory, 22(6), 644-654, 1976.

[22] M. M. E. A. Mahmoud, J. Mišić, K. Akkaya and X. Shen, *Investigating Public-Key Certificate Revocation in Smart Grid.* IEEE Internet of Things Journal, 2(6), 490-503, 2015.

[23] A. Shamir, *Identity-based Cryptosystems and Signature Schemes.* Advances in Cryptology, Crypto' 84, 1984.

[24] D. Boneh and M. Franklin, *Identity-based Encryption From the Weil Pairing*. Advances in Cryptology, 2001.

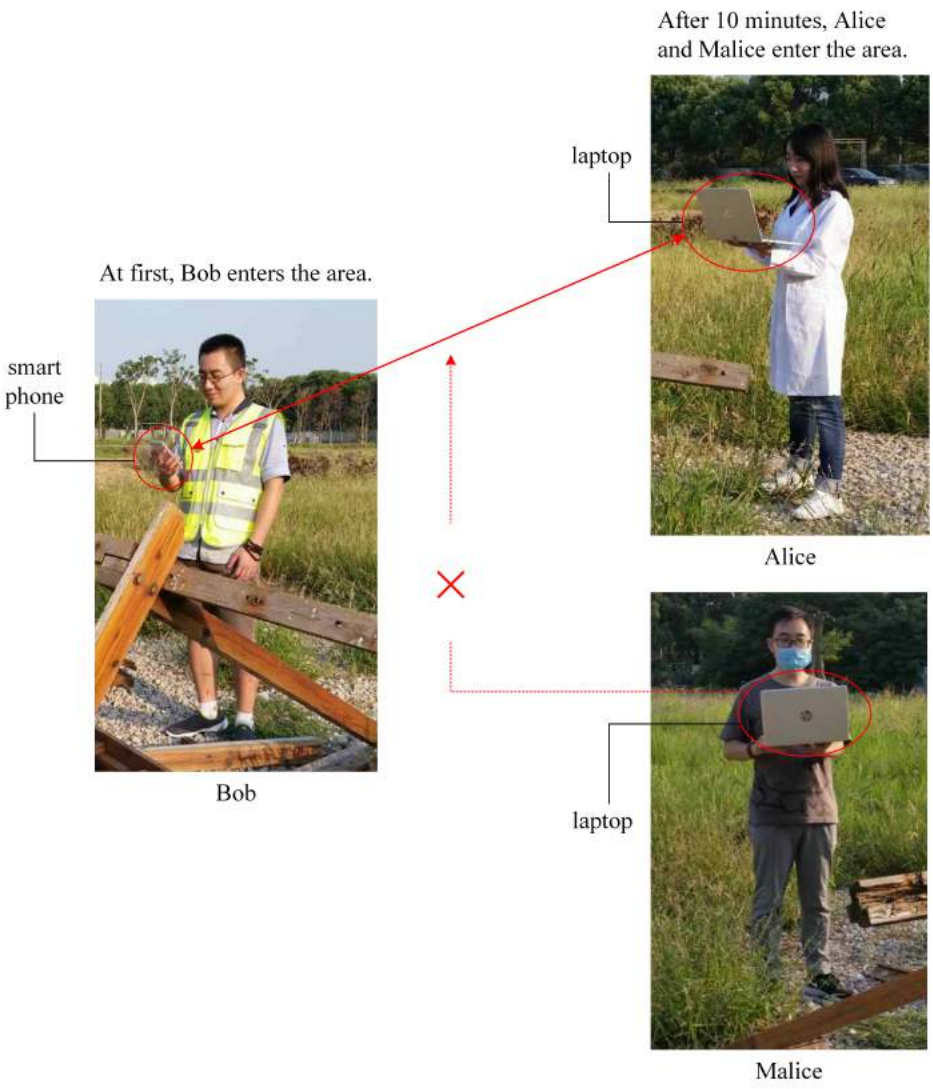

Fig. 8. Two "rescuers" and a "criminal" enter the "affected area".

[25] M. Bellare, D. Pointcheval and P. Rogaway, *Authenticated Key Exchange Secure Against Dictionary Attacks.* Advances in Cryptology C EUROCRYPT 2000, Lecture Notes in Computer Science 1807, 139–155, 2000.
[26] B. LaMacchia, K. Lauter and A. Mityagin, *Stronger Security of Authenticated Key Exchange.* International Conference on Provable Security, ProvSec 2007, Lecture Notes in Computer Science, 4784, 1–16, 2007.
[27] C. Kudla and K.G. Paterson, *Modular Security Proofs for Key Agreement Protocols.* Proceedings of the ASIACRYPT 2005, Lecture Notes in Computer Science 3788, 549C-565, 2005.
[28] G. R. Thomas, P. Armstrong, A. Boulgakov and A. W. Roscoe, *FDR3: A Modern Refinement Checker for CSP.* Tools and Algorithms for the Construction and Analysis of Systems, 8413, 187–201, 2014.
[29] C. A. R. Hoare, *Communicating Sequential Processes*. Communications of the ACM, 21(8), 666–677, 1978.
[30] R. Needham and M. D. Schroeder, *Using Encryption for Authentication in Large Networks of Computers.* Communications of the ACM, 21(12), 993–999, 1978.
[31] D. Dolev and A. Yao, *On the Security of Public Key Protocols.* Information Theory, IEEE Transactions on, 29(2), 198–208, 1983.