Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. III (2008), No. 4, pp. 395-411

McCLS: Certificateless Signature Scheme for Emergency Mobile Wireless Cyber-Physical Systems

Zhong Xu, Xue Liu, Guoqing Zhang, Wenbo He

Abstract: Mobile Ad Hoc Network is a self-configurable and self-organizing wireless network of mobile devices without fixed infrastructure support, which makes it a good candidate as underlying communication network for the Cyber-Physical Systems in emergency conditions such as earthquake, flood, and battlefields. In these scenarios, efficient communication schemes with security support are especially desired. Two cryptography approaches, the public key cryptography and the identity-based cryptography, face the costly and complex key management problem and the "key escrow" problem in the real-life deployment. Recently, the certificateless public key cryptography (CL-PKC) was introduced to address these problems in previous approaches. However, the efficiency of the schemes based on CL-PKC is not high and can be improved further.

In this paper, we present an improved certificateless signature scheme (McCLS) based on bilinear pairings. First, we theoretically compare the efficiency of McCLS with that of existing certificateless signature schemes (CLS). Second, an empirical study is conducted to compare the traditional AODV with the McCLS scheme based on AODV (McDV) in their efficiency and effectiveness against two most common attacks (i.e. *redirection attack* and *rushing attack*). Results from theoretical analysis show that the new McCLS scheme is more efficient than existing CLS solutions, and results from empirical studies show that the McDV is able to resist the two common attacks without causing substantial degradation of the network performance.

Keywords: Certificateless Signature, MANETs, Cyber-Physical Systems, Security;

1 Introduction

A salient feature of cyber-physical systems (CPS) is that it integrates computing, monitoring, and communication capabilities, and constantly interacts with the physical environment. As a result, cyber-physical system must be dependable, safe, secure and efficient [16].

Many emergency applications such as earthquake, flood and battlefields [10] proposed for CPS will be implemented on networked environments where computing devices are connected through wireless links. For many applications such as the military applications, fixed infrastructure may not be available in the environment or even be destroyed [9]. It is important to solve the connectivity problems with self-configurable and self-organizing characteristics. A possible solution for the lack of communication means is deployment of the Mobile Ad Hoc Networks (MANETs).

While MANETs provide a great flexibility for establishing communications, they are particularly prone to the security threats of eavesdropping, interception and routing attacks. Some of these problems may be solved or mitigated with the use of cryptographic schemes [7]. In the recent literature many papers make specific proposals on determining how to use Public Key Infrastructure (PKI) [27, 23, 4, 15] and Identity-Based Public Key Cryptography (ID-PKC) [20, 13, 8, 25] cryptographic techniques to secure MANETs.

The traditional PKI signature scheme uses a centralized certificate authority to issue a digital certificate that binds a user with the corresponding public key. The requirement of certificate authority inevitably leads to complex certificate management problems in practice.

The ID-PKC which was introduced by Shamir [20] is developed from traditional PKI to simplify the certificate management process. In the ID-PKC based scheme, user's public key is derived directly from certain aspects of his identity such as email address which is assumed to be publicly known. A private key is generated by a trusted third party – Private Key Generator (PKG). However, a new inherent problem is brought by this approach, namely the "key escrow" problem since the private key of user is known to the PKG. As a result, the PKG is able to impersonate any user of its choice, or decrypt messages.

In order to solve the costly and complex key management problems in PKI and the "key escrow" problem in ID-PKC respectively, Al-Riyami and Paterson [1] proposed the first Certificateless Public key Cryptography (CL-PKC) scheme. In the certificateless signature (CLS) scheme, Key Generation Center (KGC) only provides user with a partial private key, which is related to the user's identity and the master private key only known by PKC. Then the user generates the remaining part of the private key and the corresponding public key. As a result, the KGC does not know the user's private key because the user's private key is generated by user itself, thereby solving the "key escrow" problem in ID-PKC based schemes.

However, CLS schemes are usually computationally intensive, and hence they are not readily applicable in practical applications. In this paper, we present McCLS scheme, a new CLS scheme for mobile wireless cyber-physical systems.

Compared with existing CLS schemes, McCLS scheme only requires one pairing operation in the verification phase, and none in the signing phase. Since the pairing operation is the most time-consuming computation in pairing-based cryptosystems, our McCLS scheme has less computation overhead and therefore is more efficient than those schemes proposed previously in [1, 12, 14, 26]. We also provide a detailed security proof for McCLS scheme based on the Computational Diffie-Hellman Problem (*CDHP*) [6]. Then an empirical study is conducted to compare the McCLS based on AODV (McDV) with the traditional AODV in their efficiency and effectiveness against the two most common attacks, *redirection attack* and *rushing attack*, based on QualNet simulation software [19]. Results show that our scheme is efficient in terms of computation overhead and it can resist *redirection attack* [18] and *rushing attack* [11].

The remainder of the paper is organized as follows. Section 2 provides a brief description on the related work. Section 3 introduces the preliminaries and the background on the security model and the attack model. Section 4 presents our efficient McCLS scheme. Section 5 analyzes the security of McCLS scheme in detail. Section 6 evaluates the performance of McCLS scheme under the redirection attack and the rushing attack. Finally, Section 7 concludes this paper with summaries and the directions of future work.

2 Related Work

Cyber-Physical Systems (CPS) are physical and engineered systems whose operations are integrated, monitored, and controlled by a computational core [17]. CPS integrate the communication and computation with the physical process [2]. Since cyber-physical systems constantly interact with the physical environment, they must be dependable, safe, secure and efficient [16].

CPS is a new active research area. The position papers published in the NSF workshop on Cyber-Physical System [16] presents a good overview of the different aspects of CPS research. Though security is an important research issue of CPS , little work has been done [3] so far for the security of CPS.

Since many emergency applications proposed for CPS will be implemented on mobile ad hoc networks (MANETs), it is natural to ask the question if security schemes proposed for MANETs are practical for CPS. To overcome the security problems in MANETs due to their infrastructure-less nature, we need some new methods to solve these problems. One of these methods is the lightweight and efficient key management scheme. Recently, in order to solve the key management problem in public key

cryptography and “key escrow" problem in identity-based cryptography schemes, Al-Riyami and Paterson [1] proposed the first certificateless signature (CLS) scheme but fail to provide the security proof. Later, Huang et al. [12] found that this CLS scheme was insecure against a Type I forger attack. A modified CLS scheme was proposed with security proved under the random oracle model [5]. However, the scheme requires more pairing operations than the original scheme proposed in [1]. In [14], Li et al. therefore, proposed another CLS scheme, with a formal security analysis omitted. Another shortcoming of this scheme is that the verification algorithm requires four quite expensive pairing operations. Zhang et al. [26] presented a CLS scheme with a formal security analysis but it still needs four pairing operations in the verification phase. Following that, Yap et al. [22] proposed a new CLS scheme, which requires no pairing operation in the signing phase but requires two pairing operations in the verification phase.

However, since pairing operations are costly in computation and are usually time consuming, using more pairing operations in the scheme will make it difficult to be applied for emergency cyber-physical systems, because CPS need constantly interact with the physical environment and with stringent timing requirements. In this paper, we present McCLS scheme, which is more efficient and hence is a good alternative to be used in cyber-physical systems.

A good security protocol must be resilient against security attacks. In the following, we briefly introduce two most commonly studied attacks. Later in this paper, we prove that the proposed McCLS scheme is resilient against these two attacks.

Redirection attack [18] is one of the many possible attacks in MANETs. In this attack, a malicious node sends a forged Route Reply (RREP) packet to a source node by altering control message fields with falsified values. When a source node receives multiple RREPs, by comparing the destination sequence numbers contained in RREP packets, it regards the largest one as the most recent routing information and selects the route through which that RREP packet has been sent. If the attacker sends the RREP with destination sequence number higher than that of the real destination node to the source node, the data traffic will be directed toward the attacker. It then drops all data packets it receives instead of forwarding them to the next node on the routing path. Consequently, the source and destination nodes will lose communication with each other.

Rushing attack [11] usually aims at a reactive routing protocol. Every node in the network only forwards the first route discovery packet that it receives and drops the rest. Malicious nodes can “rush" the route request packets towards the destination. As a result, the nodes that receive these “rushed" request packets forward them and discard other route requests that arrive later. The resulted routes would then include the malicious nodes. In this way, the attacker is placed in an advantageous position.

3 Preliminaries

In this section, we present some mathematical background which helps in realizing CLS based on the bilinear pairing. It is commonly used in CLS schemes to realize signature and verification [1, 12, 14].

We define two cyclic groups G_1, G_2, where G_1 is an additive group and G_2 is a multiplicative group, where both groups have a prime order p. Let e be a computable bilinear map $e : G_1 \times G_1 \rightarrow G_2$. We have the following conditions:

1. Bilinearity: For any $P, Q, R \in G_1$, we have $e(P+Q,R) = e(P,R)e(Q,R)$. For $a, b \in \mathbb{Z}_p^*$ and $P, Q \in G_1$, we have $e(aP, bQ) = e(P,Q)^{ab} = e(P, abP) = e(abP, P)$.

2. Non-degeneracy: There exists $P, Q \in G_1$, such that $e(P,Q) \neq 1$.

3. Computability: There is an efficient algorithm to compute $e(P,Q)$ for all $P, Q \in G_1$

The map e will be derived from either the Weil or Tate pairing on an elliptic curve over a finite field.

An efficiently computable bilinear map e provides an algorithm for solving the Decision Diffie-Hellman Problem *(DDHP)* [6]. That is, given $(P, aP, bP, cP) \in G_1$ and $a, b \in \mathbb{Z}_p^*$, decide whether $c \equiv ab \in \mathbb{Z}_p^*$.

In bilinear pairing, Decision Diffie-Hellman (*DDH*) problem is easy and Computational Diffie-Hellman (*CDH*) problem [22] is still hard. That is, for $a, b \in \mathbb{Z}_p^*$, given (P, aP, bP), computing abP is infeasible.

3.1 Certificateless scheme

Usually, a certificateless signature (CLS) scheme consists of five polynomial time algorithms [1]:

- **Setup.** KGC runs a probabilistic algorithm to initialize the system. It receives a security parameter k and returns a randomly chosen master key and a list of public parameters **param**.
- **Extract Partial Private Key.** KGC takes the master key and an identity $ID \in \{0,1\}^*$ as inputs, and outputs a private partial key D_{ID}.
- **Generate Key Pair.** The user takes a list of public parameters **param** as inputs, outputs a private key S_{ID} and a public key P_{ID}.
- **CL-Sign.** The user takes a list of public parameters **param**, full private keys (D_{ID}, S_{ID}), and a message M to produce a signature σ on M.
- **CL-Verify.** Anyone in this algorithm may take $\{\textbf{param}, ID, P_{ID}\}$ and a message M as inputs, and outputs *true* if and only if σ is the valid signature, or a symbol $\perp$ to indicate a failure.

We may note that, once the user received the public parameters, such as public key of KGC, user chooses secret value to generate his key pair including user's private key and user's public key. Thus the user's full private key is composed of the partial private key generated by KGC and the user's private key generated by user himself. Neither the KGC nor the user can generate the full private keys by himself, therefore solving the "key escrow" problem.

3.2 Adversarial Model

As defined in [1, 22], there are two types of adversaries, Type I and Type II, with different capabilities. In CLS, Type I Adversary A_I acts as a third part who tries to impersonate a user. It is not allowed to know the KGC's master private key. However, A_I can replace the public key P_{ID} with values of its choice due to nature of the public key generated by the user. This means the adversary is able to fool the user accepting the signature, which is signed by the adversary's public key. Type II adversary A_{II} represents a malicious KGC who knows the master private key. That is, A_{II} can compute the partial private key by itself. But A_{II} does not know the user's private key S_{ID} and it cannot replace the user's public keys P_{ID} .

Definition 1. A CLS scheme is secure against existential forgery on adaptive chosen message and ID attacks against adversary A, of Type I or Type II if no polynomial time algorithm has a non-negligible advantage against a challenger C in the following game [1]:

1. The challenger C takes a security parameter k and runs the *Setup* algorithm. Challenger C gives A the system parameters **param**. If A is of Type I, the challenge C keeps the master private key to itself. Otherwise, C gives the master private key to A.
2. A can request C to answer the following types of queries:

- **Partial Key Extraction** (For Type I adversary only). C returns to A's partial private key D_{ID} as the result of running **Extract Partial Private Key** algorithm.
- **Secret Value and Public Key Extraction**. C returns to A's private key S_{ID} associated with A's public key P_{ID} as the result of running **Extract Partial Private Key** and **Generate Key Pair** algorithms. In the case of Type I adversary, C returns if the user's public key P_{ID} has been replaced.
- **Public Key Replacement** (For Type I adversary only). A can replace the associated public key P_{ID} to a new public key P'_{ID} which is chosen by itself.
- **Sign**. C returns a valid signature σ using **CL-sign** algorithm regardless whether the public key P_{ID} has been replaced or not.

3. Eventually, A outputs a signature (ID^*, m^*, σ^*). A wins game if $Verify(param, P_{ID^*}, m^*, \sigma^*) = true$ and the generated output fulfills the following conditions:

   - CL-Sign(ID^*, m^*) has never been queried.
   - If adversary A is Type I, ID^* has not been submitted to **Partial Key Extraction**.
   - If adversary A is Type II, ID^* has not been submitted to **Secret Value and Public Key Extraction**.

4 McCLS Scheme

McCLS scheme is motivated by the identity-based signature from [24]. Our verification phase algorithm requires one pairing operation only, hence McCLS scheme outperforms the other existing CLS schemes in terms of efficiency. Besides, message signing in McCLS scheme is fast as it involves no pairing computation. McCLS scheme is comprised of the following five stages.

- **Setup.** Given a cyclic group G_1 of prime order p, with an admissible pairing e and its generator P, KGC picks $s \in \mathbb{Z}_p^*$ and sets $P_{pub} = sP$. Then Chooses two hash functions $H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2 : \{0,1\}^* \times G_1 \rightarrow \mathbb{Z}_p^*$. The public system parameter list is (P, P_{pub}, H_1, H_2), and the master private key is $msk = s$.
- **Extract Partial Private Key.** Given an identity ID, KGC computes $Q_{ID} = H_1(ID)$ and $D_{ID} = sH_1(ID)$. Output D_{ID} as the partial private key corresponding to $Q_{ID} = H_1(ID)$.
- **Generate Key Pair.** The user generates a secret value $x \in \mathbb{Z}_p^*$, the public key is $P_{ID} = xP_{pub}$. The user's private key is $S_{ID} = x$.
- **Sign.** Given the user's full private keys (D_{ID}, S_{ID}) and a message M, user picks a number $r \in \mathbb{Z}_p^*$ and outputs a signature $\sigma = (V, S, R)$ where $S = \frac{1}{S_{ID}} D_{ID}$, $R = (r - S_{ID})P$ and $V = H_2(M, R, P_{ID})rP$.
- **Verification.** Given the signature (V, S, R) of a message M for the identity ID, anyone in this algorithm can act a verifier to compute $h = H_2(M, R, P_{ID})$. Then checks whether $(P_{pub}, V - hR, S/h, Q_{ID})$ is a valid Diffie-Hellman tuple, that is, computes whether $e(P_{pub}, Q_{ID}) = e(V - hR, S/h)$. If yes, accept the signature. Otherwise, reject it.

5 Analysis of McCLS Scheme

In this section, we analyze the correctness, performance and security proof of McCLS scheme.

5.1 Correctness

The correctness of McCLS scheme can be verified as follows:

$$\begin{aligned} & e(V-hR, S/h) \\ = \; & e(hrP-hrP+xhP, S/h) \\ = \; & e(xhP, D_{ID}/xh) \\ = \; & e(P_{pub}, Q_{ID}). \end{aligned}$$

Note that $e(P_{pub}, Q_{ID})$ is independent of the message, and only needs to be computed once and for all. So McCLS scheme is more efficient than other previous schemes.

5.2 Performance

McCLS scheme only requires two scalar multiplication in signature phase and two scalar multiplication computations and one pairing operation in verification phase. The pairing operations are expensive comparing with scalar multiplication and exponentiation.

The comparison between the exiting schemes and McCLS scheme according to efficiency of sign and verification algorithms and the length of public keys is shown in Table 1. It shows that McCLS scheme has the lowest pairing operations requirement and has the same length of public key as other CLS schemes.

Table 1: Comparison of the CLS Schemes

	AP [1]	LCS [14]	ZWXF [26]	YHG [22]	McCLS
Sign	1p+3s	2s	3s	2s	2s
Verify	4p+1e	4p+2s	4p	2p+3s	1p+3s
Pklen	2 points	2 points	1 points	1 point	1 point

Pklen: the public key length;
s: the scalar multiplication computation;
p: the pairing operation;
e: the exponential computation.

5.3 Security Proof

In this section we discuss the security of McCLS scheme under the security model discussed in section 3. The main theorems concerning the security of our scheme are:

Theorem 2. *Our certificateless signature scheme is existentially unforgeable against a Type I adversary A_I in the random oracle model under the assumption that the CDH problem in G_1 is infeasible.*

Proof. Suppose there exists an adversary A_I which has an advantage in attacking McCLS scheme. We build a challenger C that uses A_I to solve the *CDH* problem. C receives an instance (P, aP, bP) of the *CDHP*. Its goal is to compute abP. On the setup phase, C sets P as the generator of the group, and sets $P_{pub} = aP$ where a is the master key, which is unknown to A_I. In order to avoid collision, C maintains a list $L = (ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$ throughout the game. The list is initially empty. C then starts to answer oracle queries with the following procedures [26]:

- H_1 **Queries**. Suppose A_I makes q_{H_1} queries to H_1 oracle, where q_{H_1} denotes the maximum number of queries. Randomly choose $j \in [1, q_{H_1}]$. When an identity ID_i is submitted to oracle H_1 where $i \in [1, q_{H_1}]$, if $i = j$, assume that $ID_i = ID^*$ at this point, C saves a list $L_1 = (ID_i, Q_i, y_i)$ where $Q_i = bP$, $y_i = \perp$ (indicate to failure). Otherwise, C generates a random number y_i and lets $Q_i = y_i P$, then saves $L_1 = (ID_i, Q_i, y_i)$.

- **Partial Key Extraction** (ID_i) **Queries**. When A_I makes the query on ID_i, if $ID_i = ID^*$, then C aborts and halts the simulation. Otherwise C finds L and performs as follows:
  - If the list L contains $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C checks whether $D_{ID_i} = \perp$. If $D_{ID_i} \neq \perp$, C returns D_{ID_i} to A_I. If $D_{ID_i} = \perp$, and $ID_i \neq ID^*$, C answers with $D_{ID_i} = y_i P_{pub} = y_i(aP)$ as partial private key. C then returns D_{ID_i} to A_I and adds it to L.
  - If the list L does not contain $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C sets $(D_{ID_i} = y_i P_{pub} = y_i(aP))$. Then challenger C sets $(s_{ID}, P_{ID}) = \perp$ and adds $ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i}$ to the list L.

- **Public Key Extraction** (ID_i) **Queries**. When A_I makes the query on ID_i, C finds L and performs as follows:
  - If the list L contains $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C checks whether $P_{ID_i} = \perp$. If $P_{ID_i} \neq \perp$, C returns P_{ID_i} to A_I. Otherwise, C picks a random $x_i \in \mathbb{Z}_p^*$, and sets $P_{ID_i} = x_i P_{pub}$, $s_{ID_i} = x_i$. C then returns P_{ID_i} to A_I and adds (s_{ID_i}, P_{ID_i}) to L.
  - If the list L does not contain $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C picks a random $x_i \in \mathbb{Z}_p^*$, and sets $P_{ID_i} = x_i P_{pub}$, $s_{ID_i} = x_i$. C then returns P_{ID_i} to A_I and adds (s_{ID_i}, P_{ID_i}) to L.

- **Secret Value Extraction** (ID_i) **Queries**. When A_I makes the query on ID_i, if $ID_i = ID^*$, then C aborts and halts the simulation. Otherwise C finds L and performs as follows:
  - If the list L contains $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C checks whether $D_{ID_i} = \perp$. If $D_{ID_i} = \perp$, C executes **Partial Key Extraction Queries** to obtain D_{ID_i}. If $P_{ID_i} = \perp$, C makes **Public Key Extraction Queries** to obtain $s_{ID_i} = x_i, P_{ID_i} = x_i P_{pub}$. Then C saves the value and adds full private keys (D_{ID_i}, s_{ID_i}) to the list L.
  - If the list L does not contain $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C executes **Partial Key Extraction Queries** to obtain $D_{ID_i}$and makes **Public Key Extraction Queries** to obtain (s_{ID_i}, P_{ID_i}). Then C saves the value and adds full private keys (D_{ID_i}, s_{ID_i}) to the list L.

- **Public Key Replacement**(ID_i, P'_{ID}) **Queries**. When A_I makes the query on (ID_i, P'_{ID_i}), C finds L and performs as follows:
  - If the list L contains $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C sets $P_{ID_i} = P_{ID'_i}$ and $s_{ID_i} = \perp$.
  - If the list L does not contain $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C sets $D_{ID_i} = \perp$, $P_{ID_i} = P_{ID'_i}$ and $s_{ID_i} = \perp$. And then C adds to the list L.

- H_2 **Queries**. When A_I makes the query on (m, R, P_{ID_i}), C first scans if a list $L_2 = (m, R, P_{ID_i}, h_j)$ has been defined. If defined, return the list to A_I. Otherwise, C picks a random $h_j \in \mathbb{Z}_p^*$ as the hash value and returns h_j, and adds it to L_2.

- **Sign Queries**(ID_i, M_j). When A_I asks for a signature by user ID_i on message M_j. C finds $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$. If D_{ID_i} not found, C runs **Partial Key Extraction Queries**. If (P_{ID_i}, s_{ID_i}) not found, C runs **Public Key Extraction Queries**. Note that if $ID_i \neq ID^*$, A_I is able to generate signature on any messages using corresponding full private keys (D_{ID_i}, s_{ID_i}). As far as $ID_i = ID^*$, assume that P_{ID_i} is current public key and corresponding private key $s_{ID_i} = x$, where $x \in \mathbb{Z}_p^*$, additionally submits through the A_I. This is because the public key has been replaced earlier by A_I, then C cannot know the corresponding private key and thus the signing oracle's answer may not be correct.

  On receiving sign queries, C does the following:

1. Choose random $r_j \in \mathbb{Z}_p^*$ and look up the list L_2 for h_j, if not found, C runs H_2 **Queries** to get h_j.
2. Compute $V_j = h_j(x + \frac{a}{r_j})P$ and $S_j = r_j Q_i = r_j bP, R_j = xP$;
3. Return the signature $\sigma = (V_j, S_j, R_j)$.

Now, σ is returned to A_I, which appears to be valid signature since

$$\begin{aligned} & e(V_j - h_j R, S_j/h_j) \\ = \; & e(h_j(x+\tfrac{a}{r_j})P) - h_j xP, r_j bP/h_j) \\ = \; & e(h_j aP/r_j, r_j bP/h_j) \\ = \; & e(aP, bP) \\ = \; & e(P_{pub}, Q_{ID}). \end{aligned}$$

Finally, A_I will output a valid forgery $r = (ID_j, M_j, R_j, S_j, V_j)$. If $ID_j \neq ID^*$, C outputs the FAIL and aborts the simulation. Otherwise, we can compute r_j through $r_j = \frac{ah_j}{V_j - h_j x}$ [21], since $(P_{pub}, V_j P - h_j R_j, S_j/h_j, Q_i)$ is a valid Diffie-Hellman tuple. Apply r_j to S_j, we have

$$S_j = \frac{ah_j}{V_j - h_j x} Q_i$$

$$S_j = \frac{ah_j}{V_j - h_j x} bP$$

$$abP = S_j(V_j - h_j x)/h_j \quad .$$

So $abP = S_j(V_j - h_j x)/h_j$ is the answer to our *CDHP* instance. If the A_I can break our scheme, then the attacker solves the *CDH* problem.

Theorem 3. *Our certificateless signature scheme is existentially unforgeable against the A_{II} adversary in the random oracle model under the assumption that the CDH problem in G_1 is infeasible.*

Proof. Suppose there exists an adversary A_{II} which has advantage in attacking McCLS scheme. We build a challenger C that uses A_{II} to solve the *CDH* problem. C receives an instance (P, aP, bP) of the *CDHP*. Its goal is to compute abP. On the setup phase, C sets P as the generator of the group, and sets $P_{pub} = sP$ where s is the master key, which is known to A_{II}. In order to avoid collision, C maintains a list $L = (ID_i, s_{ID_i}, P_{ID_i})$ throughout the game. The list is initially empty. C then starts to answer oracle queries with the following procedures:

- H_1 **Queries**. Suppose A_{II} makes q_{H_1} queries to H_1 oracle, where q_{H_1} denotes the maximum number of queries. Randomly choose $j \in [1, q_{H_1}]$. When an identity ID_i is submitted to oracle H_1 where $i \in [1, q_{H_1}]$, if $i = j$, assume that $ID_i = ID^*$ at this point, C saves a list $L_1 = (ID_i, Q_i, y_i)$ where $Q_i = aP$, $y_i = \perp$ (indicate to failure). Otherwise, C generates a random number y_i and lets $Q_i = y_i P$, and saves $L_1 = (ID_i, Q_i, y_i)$.
- **Public Key Extraction** (ID_i) **Queries**. When A_{II} makes the query on ID_i, C finds L and performs as follows:
  - If the list L contains $(ID_i, s_{ID_i}, P_{ID_i})$, C checks whether $P_{ID_i} = \perp$. If $P_{ID_i} \neq \perp$, C returns P_{ID_i} to A_{II}. Otherwise, C picks a random $x_i \in \mathbb{Z}_p^*$, and sets $P_{ID_i} = bP_{pub}$, $s_{ID_i} = x_i$. C then returns P_{ID_i} to A_{II} and adds (s_{ID_i}, P_{ID_i}) to L.
  - If the list L does not contain $(ID_i, s_{ID_i}, P_{ID_i})$, C picks a random $x_i \in \mathbb{Z}_p^*$, and sets $P_{ID_i} = bP_{pub}$, $s_{ID_i} = x_i$. C then returns P_{ID_i} to A_{II} and adds (s_{ID_i}, P_{ID_i}) to L.

- **Secret Value Extraction** (ID_i) **Queries**. When A_{II} makes the query on ID_i, if $ID_i = ID^*$, then C aborts and halts the simulation. Otherwise C finds L and performs as follows:

  - If the list L contains $(ID_i, s_{ID_i}, P_{ID_i})$, C checks whether $P_{ID_i} = \perp$. If $P_{ID_i} = \perp$, C makes **Public Key Extraction Queries** to obtain $(s_{ID_i} = x_i, P_{ID_i} = x_i P_{pub})$. Then C saves the value and adds user's private keys s_{ID_i} to the list L.
  - If the list L does not contain $(ID_i, s_{ID_i}, P_{ID_i})$, C executes **Public Key Extraction Queries** to obtain (s_{ID_i}, P_{ID_i}). Then C saves the value and adds user's private keys s_{ID_i} to the list L.

- H_2 **Queries**. When A_{II} makes the query on (m, R, P_{ID_i}), C first scans whether a list $L_2 = (m, R, P_{ID_i}, h_j)$ has been defined. If defined, return the list to A_{II}. otherwise, C picks a random $h_j \in \mathbb{Z}_p^*$ as the hash value of Hand returns h_j, and adds it to L_2.

- **Sign Queries**(ID_i, M_j). When A_{II} asks for a signature by user ID_i on message M_j. C finds $(ID_i, s_{ID_i}, P_{ID_i})$. If (P_{ID_i}, s_{ID_i}) not found, C runs **Public Key Extraction Queries**.

  On receiving sign queries, C does the following:

  1. Choose random $r_j \in \mathbb{Z}_p^*$ and look up the list L_2 for h_j, if not found, C runs H_2 **Queries** to get h_j.
  2. Compute $V_j = (\frac{sh_j + bh_j}{r_j x_i})P$ and $S_j = r_j x_i Q_i = r_j x_i aP, R_j = \frac{bP}{r_j x_i}$;
  3. Return the signature $\sigma = (V_j, S_j, R_j)$.

  Now, σ is returned to A_{II}, which appears to be valid signature since

$$\begin{aligned} & e(V_j - h_j R, S_j / h_j) \\ = \; & e((\tfrac{sh_j + bh_j}{r_j x_i} P) - h_j \tfrac{bP}{r_j x_i}, r_j x_i aP / h_j) \\ = \; & e(\tfrac{sPh_j}{r_j x_i}, r_j x_i aP / h_j) \\ = \; & e(sP, aP) \\ = \; & e(P_{pub}, Q_{ID}). \end{aligned}$$

Finally, A_{II} will output a valid forgery $r = (ID_j, M_j, R_j, S_j, V_j)$. If $ID_j \neq ID^*$, C outputs the FAIL and aborts the simulation. Otherwise, we can compute r through $r_j = \frac{sh_j + bh_j}{xV_j}$, since $(P_{pub}, V_j P - h_j R_j, S_j / h_j, Q_i)$ is a valid Diffie-Hellman tuple. Apply r_j to S_j, we have

$$S_j = \frac{sh_j + bh_j}{x_i V_j} x_i Q_i$$

$$S_j = \frac{Q_i sh_j + bh_j aP}{V}$$

$$abP = \frac{V_j r_j - Q_i S_j h_j}{h_j} \quad .$$

So $abP = \frac{V_j r_j - Q_i S_j h_j}{h_j}$ is the answer to our *CDHP* instance. If the A_{II} can break our scheme, then the attacker solves the *CDH* problem.

Table 2: General parameters

Parameter	Value
Transmitter	250m
Bandwidth	2Mb/s
Simulation time	600s
Environment	900m×900m
Traffic type	CBR (Constant Bit Rate)
Packet rate	4 packets/s
Packet size	512 bytes
Node maximum speed	0, 5,10,15,20 m/s
Pause time	0s
Attack nodes	1,2 and 4 Redirection, 1,2 and 4 Rushing
Queuing policy at routers	First-in-first-out

6 Evaluation and Analysis

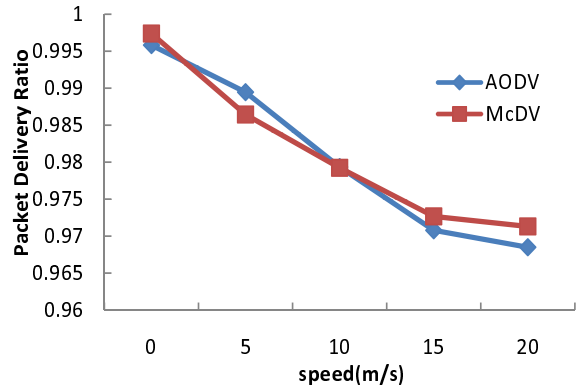
In this section, an efficient McCLS scheme named McDV based on the Ad hoc On-Demand Distance Vector Routing (AODV) is proposed. We start the simulations using QualNet [19] in order to compare the original AODV protocol without any security requirements with McDV based on the CLS with routing authentication extension. We also evaluate the performance of two schemes under 1, 2 and 4 nodes *redirection attacks* and 1, 2 and 4 nodes *rushing attacks*, as this is more realistic in the real emergency applications. Our implementation retains most of the AODV mechanisms, such as route discovery, reverse path setup, forwarding path setup, route maintenance, and so on. In our experiments, 20 nodes move around in a rectangular area of 900×900m according to a mobility model, i.e., the random waypoint model. The nodes spread randomly over the network. Each node starts its journey from a random location to a random destination. We vary the nodes speed from 0m/s to 20m/s, and set the nodes pause time as 0s. Table 2 lists the values of the common parameters used in all experiment. Other parameters will be given in the description of each specific experiment.

The performance of McDV is compared using the following performance metrics.

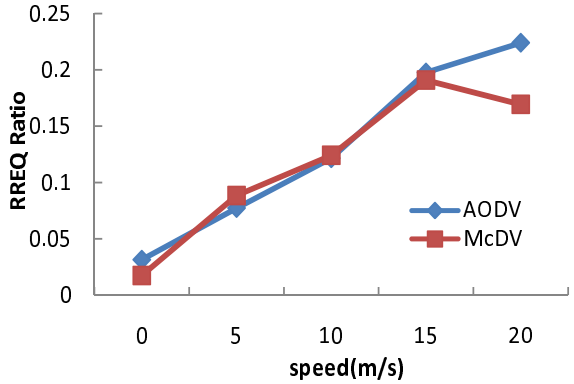
- Packet Delivery Ratio: Ratio of the number of packets received by the destination over the number of packets sent by the source.
- RREQ Ratio: Ratio of sum number of RREQ initiated, forwarded and retried over the sum of number of data packets sent as source and data packets forwarded. Present the number of RREQ packets transmitted through the network.
- End-to-End Delay: The average time experienced by each packet when traveling from the source to the destination.
- Throughput: Ratio of the total bytes sent by all sources nodes over the total time.
- Packet Drop Ratio: Ratio of the number of packets discarded by attacking nodes over the total number of packets sent by all sources.

Effects of various metrics on different protocols: Experiments in this section are used to study the performance between McDV and AODV. The results are shown in Fig. 1.

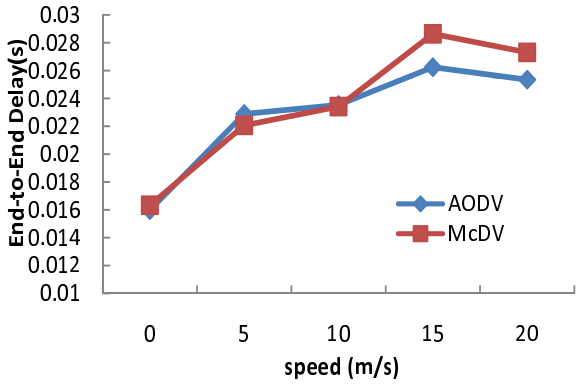
The packet delivery ratio and the RREQ ratio are shown in Fig. 1(a) and Fig. 1(b), respectively. We can see that McDV could work well in the experiment because the packet delivery ratio and RREQ ratio in AODV are very similar to that of McDV, without causing any substantial degradation of the network



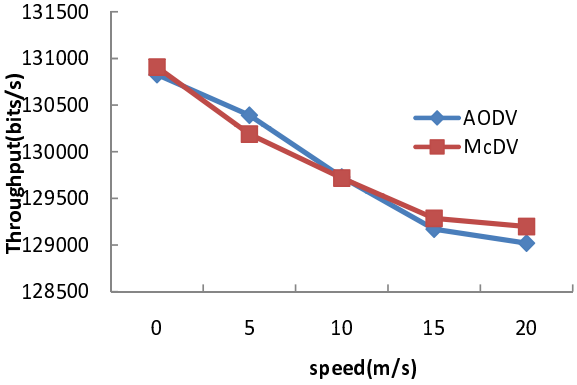
(a) Comparison with AODV and McDV, Packet Delivery Ratio



(b) Comparison with AODV and McDV, RREQ Ratio



(c) Comparison with AODV and McDV, End-to-End Delay(s)



(d) Comparison with AODV and McDV, Throughput(bit/s)

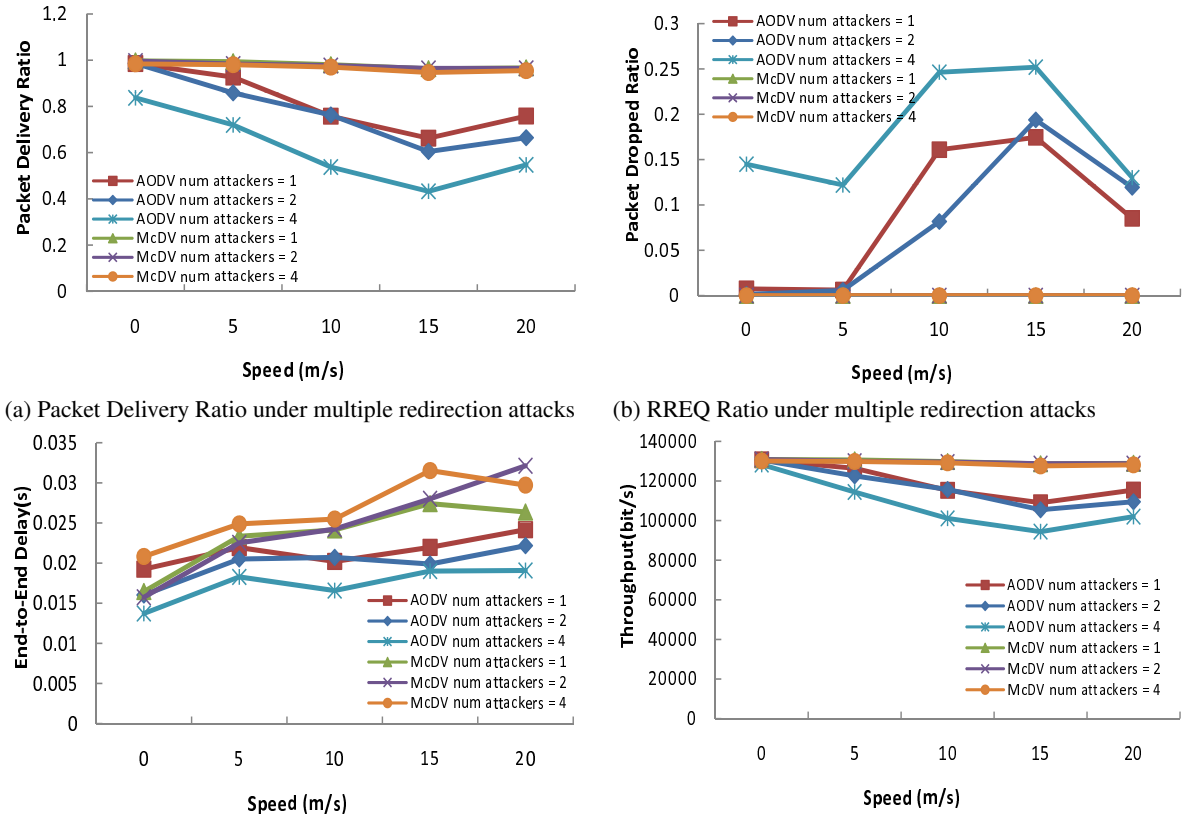
Figure 1: Effects of various metrics on different protocols

performance. As nodes speed increases, the number of data packets reaching the destination decreases and the number of RREQ packets transmitting through the networks increases.

End-to-end delay of McDV scheme is shown in Fig. 1(c). Our scheme has a little bit higher delay than that of AODV due to the exchange of packets during authentication phase of the security process. Result shows that McDV has a similar end-to-end delay with AODV at a relatively low speed, however, when the maximum speed of nodes is higher than 15m/s, AODV outperforms McDV scheme. More specifically, our scheme needs authentication operation, and those additional operations are computed in our scheme but not in AODV. We only measure delays for data packets that survived to reach their destination.

Throughput of McDV works well as result shown in Fig. 1(d) because the effect of throughput of the network is very small (around 0.16%). However, if this scheme in other real scenarios such as disaster scenarios, battlefield scenarios, or even very high-speed scenarios, the effect of throughput of the network may reduce more than this.

Effects of multiple attackers with redirection attacks: We simulated AODV and McDV under redirection attacks by varying the nodes speed from 0m/s to 20m/s while setting the number of attack nodes to 1, 2 and 4 nodes, respectively. We first study the packet delivery ratio and packet dropped ratio. From the results of Fig. 2(a), we can see that packet delivery ratio drops as the speed increases when we use AODV routing protocol under redirection attacks. Meanwhile, we observe that given the same speed of nodes, the higher the number of attackers, the lower the packet delivery ratio in AODV. The packet delivery ratio in the case of 4 attackers declines dramatically to 43% as the speed of nodes



(a) Packet Delivery Ratio under multiple redirection attacks

(b) RREQ Ratio under multiple redirection attacks

(c) End-to-End Delay under multiple redirection attacks

(d) Throughput under multiple redirection attacks

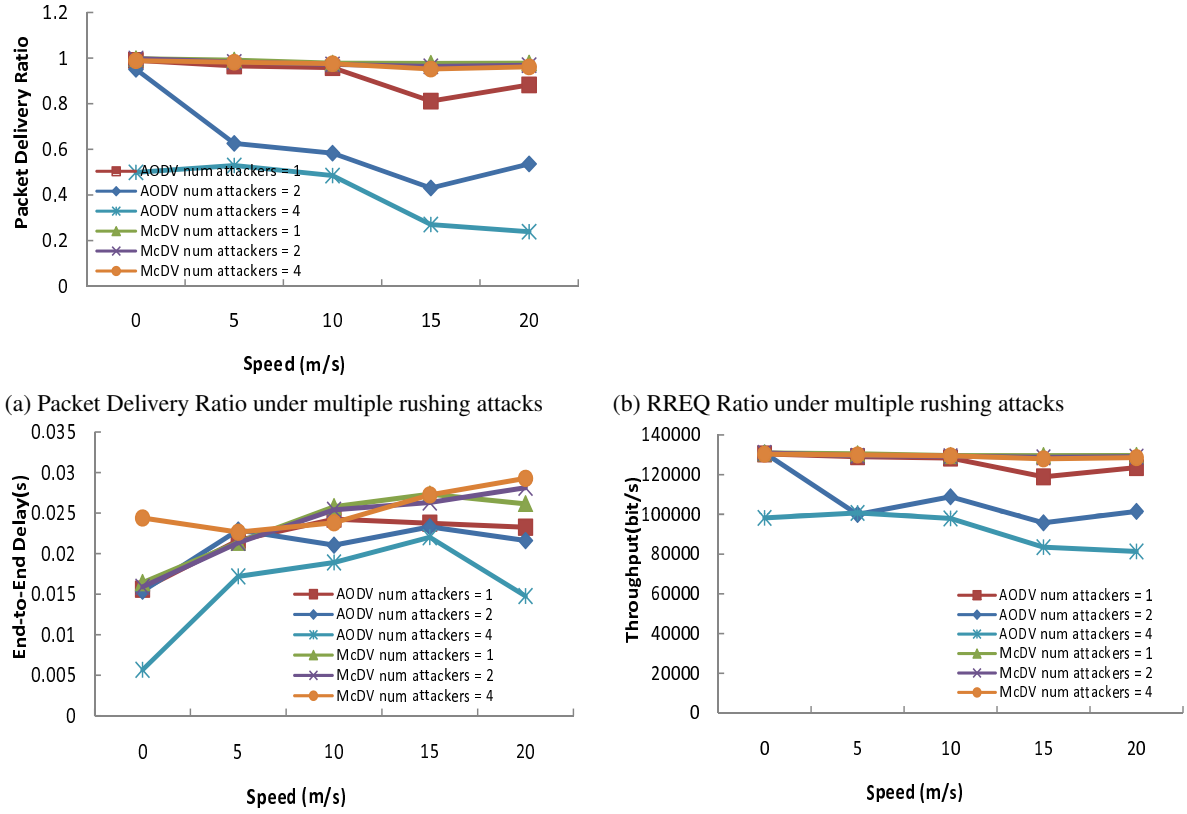
Figure 2: Effects of multiple attackers with redirection attacks

increases to 15m/s. In contrast, the packet delivery ratio of McDV maintains from 94% to 98% even the number of attackers increases to 4 which is slightly lower than normally packet delivery ratio as we can see in Fig. 1(a). All of these are brought by the fact that our routing scheme retains most of the AODV's mechanisms and the extra operations of sign phase and verification phase are very low.

As we would expect from Fig. 2(b), McDV is able to detect all redirection attacks and the packet dropped ratio is zero. On the contrary, as the attack nodes increase, the packet dropped ratio also rises at the same speed when using AODV. Especially, the highest packet dropped ratio of AODV is almost 25% at speed of 15m/s. McDV can detect all the attacks because the node will verify the signature when it receives the packet. Only if this packet passes the verification, the receiving node updates its routing table entry according to the information carried in the packet. Otherwise, the node will drop this packet.

Readers may note that in Fig. 2(c), given the same speed of nodes, the end-to-end Delay in the McDV under redirection attacks are slightly higher than the end-to-end delay in the AODV under redirection attacks. This is simply due to our definition of the end-to-end delay, which is defined as the time a packet takes to travel from the source to the destination. Given the same network size, the same number of senders and the same number of receivers, as attacker or more attackers are added to the network, the number of available nodes forwarding packet decreases, making the average end-to-end delay decrease.

The result in Fig. 2(d) shows the throughput in the network. We can see that the higher the attackers, the lower the throughput at the same speed in AODV. As the speed goes up in AODV, the throughput of network decreases. When the speed is 15m/s, the throughput of AODV drops to 76% comparing with that of original protocol. In contrast, our scheme has the similar trend as the original AODV protocol. As the speed is 15m/s and the network is under 4 redirection attackers, the most effect of throughput is



(a) Packet Delivery Ratio under multiple rushing attacks

(b) RREQ Ratio under multiple rushing attacks

(c) End-to-End Delay under multiple rushing attacks

(d) Throughput under multiple rushing attacks

Figure 3: Effects of multiple attackers with rushing attacks

around 0.9%.

Effects of multiple attackers with rushing attacks: In this section, we compare the varying metrics of AODV and that of McDV under 1, 2 and 4 rushing attacks, respectively. The graph in Fig. 3(a) shows that, the higher the nodes speed, the lower the packet delivery ratio is when using AODV. However, the packer delivery ratio declines dramatically to 24% as the number of attackers increases to 4 nodes and at the speed of 20m/s. On the other hand, the lowest packet delivery ratio in McDV still maintains 95% when the nodes are at the speed of 15m/s. The Fig. 3(b) shows that, given the same speed, the higher the attacker node(s), the higher the packet dropped ratio is under AODV. In contrast, McDV can detect all the rushing attacks, thus the packet dropped ratio is zero.

These results indicate that the AODV protocol performs worse under the rushing attacks than under the redirection attacks. This is because we set the transmit distance as 740m to simulate the rushing attacks. In this situation, the malicious nodes may readily access to the forwarding group and discard all data packets. With the number of attackers increasing, the packet delivery ratio decreases and the packet dropped ratio rises. In contrast, McDV maintains high packet delivery ratio and the packet dropped ratio is zero. This is due to its less computation overhead and efficient implementation of signature and verification.

The Fig. 3(c) shows that the end-to-end delay in rushing attacks. McDV end-to-end delay is slightly higher than AODV end-to-end delay. The explanation for this is similar to the situation discussed in the case of redirection attacks. The difference is that when a node is converted to attacker, the probability of this attacker being selected into the forwarding group increases, and the average end-to-end delay decreases.

Fig. 3(d) shows the throughput of two protocols under rushing attacks. Although mechanisms of redirection attacks and rushing attacks are different, they have a similar way to affect the throughput. The throughput drops more severely under rushing attacks than under redirection attacks. In particular, the lowest throughput almost drops to 63% under 4 rushing attacks when nodes at the speed of 20m/s. In contrast to the AODV protocol under rushing attacks, our scheme has very similar throughput to the original protocol.

7 Conclusion

An efficient certificateless signature scheme named McCLS is proposed in this paper. This scheme is based on the bilinear Diffie-Hellman assumption in the random oracle model for emergency mobile wireless cyber-physical systems. Since McCLS only requires one pairing operation in the verification phase, and none in the signing phase, theoretically it is more efficient than existing certificateless signature schemes. We also present simulation of McDV which is based on McCLS scheme and compare its performance under two most common attacks (i.e. *redirection attack* and *rushing attack*) with typical protocol-AODV providing no protection mechanism. These results show that McDV can completely resist the two kinds of attacks without causing substantial degradation of network performance. In the future, we will further investigate security schemes in the wide physical environment. Thereby we can find schemes which either prevent more comprehensive external attacks or resist internal attacks from the compromised nodes.

8 Acknowledgment

This work was supported in part by an NSERC discovery grant 341823-07 and a National Study-Abroad Scholarship of P.R.China under Grant No. [2007] 3020. Part of this work has been published in preliminary form in the proceedings of The First International Workshop on Cyber-Physical Systems, in conjunction with ICDCS 2008, Beijing, China.

Bibliography

[1] S. S. Al-Riyami and K. G.Paterson. Certificateless Public Key Cryptography. In *ASIACRYPT: Advances in Cryptology – ASIACRYPT: International Conference on the Theory and Application of Cryptology*. LNCS, Springer-Verlag, 2003.

[2] E. A.Lee. Cyber-Physical Systems - Are Computing Foundations Adequate. Technical report, UC Berkeley, 2006.

[3] M. Anand, E. Cronin, and M. Sherr. Security Challenges in Next Generation Cyber Physical Systems. Technical report, University of Pennsylvania, 2007. http://www.truststc.org/scada/papers/paper33.pdf.

[4] M. Bechler, H.-J. Hof, D. Kraft, F. Pahlke, and L.Wolf. A Cluster-based Security Architecture for Ad Hoc Networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2393–2403 vol.4, 7-11 March 2004.

[5] M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[6] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In *Advances in Cryptology - CRYPTO 2001: 21st Annual International Cryptology Conference*, volume 2139, pages 213–229. LNCS, 2001.

[7] V. Daza, J. Herranz, P. Morillo, and Carla. Cryptographic techniques for mobile ad-hoc networks. *Comput. Networks*, 51(18):4938–4950, 2007.

[8] H. Deng, A. Mukherjee, and D. P. Agrawal. Threshold and Identity-Based Key Management and Authentication for Wireless Ad Hoc Networks. In *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004*, pages 107– 111, April 2004.

[9] B. D.Noble and J. Flinm. Wireless,Self-organizing Cyber-physical systems. Technical report, University of Michigan, 2006. http://varma.ece.cmu.edu/cps/Position-Papers/Noble-Flinn.pdf.

[10] W. He, Y. Huang, K. Nahrstedt, and W. C. Lee. Smock: A self-contained public key management scheme for mission-critical wireless ad hoc networks. In *PERCOM '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications*, pages 201–210, Washington, DC, USA, 2007. IEEE Computer Society.

[11] Y.-C. Hu, A. Perrig, and D. B. Johnson. Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols. In *Proc of the ACM Workshop on Wireless Security (WiSe 2003)*, pages 30–40, 2003.

[12] X. Huang, W. Susilo, Y. Mu, and F. Zhang. On the Security of Certificateless Signature Schemes from Asiacrypt 2003. In *International Conference on Cryptology and Network Security (CANS), LNCS*, volume 4, 2005.

[13] A. Khalili, J. Katz, and W. Arbaugh. Toward Secure Key Distribution in Truly Ad Hoc Networks. In *Proc. IEEE Workshop Security and Assurance in Ad Hoc Networks*, pages 342– 346, Jan 2003.

[14] X. Li, K. Chen, and L. Sun. Certificateless Signature and Proxy Signature Schemes from Bilinear Pairings. *Lithuanian Mathematical Journal*, 45(1), 2005.

[15] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang. URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks. *IEEE/ACM Transactions on Networking*, 12(6):1049–1063, 2004.

[16] National Science Foundation. Cyber-physical systems. Technical report, NSF Workshop on Cyber-Physical Systems, 2006. http://varma.ece.cmu.edu/cps/.

[17] National Science Foundation. Computer systems research. Technical report, NSF, 2007. http://www.nsf.gov/pubs/2007/nsf07504/nsf07504.htm.

[18] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer. A secure routing protocol for ad hoc networks. *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*, pages 78–87, 12-15 Nov. 2002.

[19] Scalable Network Technologies. QualNet Simulator. http://www.scalable-networks.com/.

[20] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO: Proceedings of Crypto*, 1984.

[21] S. Xu, Y. Mu, and W. Susilo. Online/Offline Signatures and Multisignatures for AODV and DSR Routing Security. In *11th Australasian Conference on Information Security and Privacy,ACISP 2006*. LNCS, 2006.

[22] W.-S. Yap, S.-H. Heng, and B.-M. Goi. An efficient certificateless signature scheme. In *EUC Workshops*, volume 4097 of *Lecture Notes in Computer Science*, pages 322–331, 2006.

[23] S. Yi and R. Kravets. Moca: Mobile Certificate Authority for Wireless Ad Hoc Networks. In *Proc. Second Ann. PKI Research Workshop (PKI '03)*, Apr 2003.

[24] H. Yoon, J. H. Cheon, and Y. Kim. Batch Verifications with ID-Based Signatures. In *ICISC: International Conference on Information Security and Cryptology*. LNCS, 2004.

[25] Y. Zhang, W. Liu, W. Lou, Y. Fang, and Y. Kwon. AC-PKI: Anonymous and Certificateless Public-key Infrastructure for Mobile Ad Hoc Networks. In *2005 IEEE International Conference on Communications, 2005. ICC 2005*, pages 3515–3519, May 2005.

[26] Z. Zhang, D. S. Wong, J. Xu, and D. Feng. Certificateless Public-Key Signature: Security Model and Efficient Construction. In *Applied Cryptography and Network Security, 4th International Conference, ACNS 2006, Singapore, June 6-9, 2006, Proceedings*, volume 3989 of *Lecture Notes in Computer Science*, pages 293–308, 2006.

[27] L. Zhou and Z. Haas. Securing Ad Hoc Networks. *Network, IEEE*, 13(6):24–30, Nov/Dec 1999.

Zhong Xu[1,2], Xue Liu
[1]McGill University
School of Computer Science
3480 University Street, Montreal, Quebec, Canada, H3A 2A7
E-mail: {zhongxu,xueliu}@cs.mcgill.ca

Guoqing Zhang
[2]Northwestern Polytechnical University
College of Automation
Xi'an, Shaanxi, China
E-mail: gqzhang@cs.mcgill.ca

Wenbo He
University of Illinois at Urbana-Champaign
Dept. of Computer Science
Urbana, IL, USA.
E-mail: wenbohe@uiuc.edu

Zhong Xu received the B.E Degree in Automation from Xi'an Technological University in 2001 and the M.E Degree in Computer Science from XiDian University in 2005. Currently, Zhong is a joint Ph.D student in McGill University, Montreal, Canada and Northwestern Polytechnical University, Xi'an, China. From August 2001 to August 2002, he was an assistant lecturer in Xi'an Technological University, China. His research interests include Security of Ad Hoc Networks, Information Security, Embedded Systems and Cyber-Physical Systems.

Dr. Xue (Steve) Liu is an Assistant Professor in the School of Computer Science at McGill University. He is also affiliated to the Centre for Intelligent Machines (CIM). Xue obtained his Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 2006. He obtained his B.S. degree in Mathematics and M.S. degree in Automatic Control both from Tsinghua University, China. He worked briefly in the Hewlett-Packard LabsIBM T. J. Watson Research Center. He received the Ray Ozzie Fellowship, the Saburo Muroga Fellowship, the Mavis Memorial Fund Award, and the C. W. Gear Outstanding Graduate Award, from the University of Illinois at Urbana-Champaign. He has filed 5 patents, and published more than 50 research papers in international journals and major peer-reviewed conference proceedings.

Guoqing Zhang received the B.E Degree and M.E. degree in Automation both from Northwestern Polytechnical University. Currently, he is a Ph.D student in Northwestern Polytechnical University, Xi'an, China. His research interests include Vehicular Ad-hoc Networks, Information Security and Embedded Systems.

Wenbo He is currently a Ph.D student at Department of Computer Science, in University of Illinois at Urbana-Champaign, where she is advised by Professor Klara Nahrstedt. She received the Mavis Memorial Fund Scholarship Award from College of Engineering of UIUC in 2006, and C. W. Gear Outstanding Graduate Award from Department of Computer Science in 2007. She is also a recipient of Vodafone Fellowship in 2005-2008. Wenbo received the M.S. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 2000. She received the M.Eng. degree in automatic control theory from Tsinghua University, Beijing, China, in 1998, and the B.E. degree in automatic control from the Harbin Engineering University, Heilongjiang, China, in 1995. From August 2001 to January 2005, she was a Software Engineer with Cisco Systems Inc., Champaign, IL. Her research interests include pervasive and mobile computing, and network security and privacy.