Efficient Multi-Party Digital Signature using Adaptive Secret Sharing for Low-Power Devices in Wireless Networks

Caimu Tang, *Member, IEEE,* Dapeng Oliver Wu, *Senior Member, IEEE,*
Anthony T. Chronopoulos, *Senior Member, IEEE,* and Cauligi S. Raghavendra, *Fellow, IEEE*

***Abstract*—In this paper, we propose an efficient multi-party signature scheme for wireless networks where a given number of signees can jointly sign a document, and it can be verified by any entity who possesses the certified group public key. Our scheme is based on an efficient threshold key generation scheme which is able to defend against both static and adaptive adversaries. Specifically, our key generation method employs the bit commitment technique to achieve efficiency in key generation and share refreshing; our share refreshing method provides proactive protection to long-lasting secret and allows a new signee to join a signing group. We demonstrate that previous known approaches are not efficient in wireless networks, and the proposed multi-party signature scheme is flexible, efficient, and achieves strong security for low-power devices in wireless networks.**

***Index Terms*—Multi-party signature, distributed key generation, elliptic curve cryptosystems.**

I. Introduction

KEYS generated by distributed key generation (DKG) protocols [1] can be used to facilitate multi-party digital signature [1], [2], and it provides great flexibility since not all of designated signees are required to actually perform the signing process. This type of signature schemes should be efficient to be used in low-power devices since smart-cards are the widely accepted implementation platforms for many security functionalities, and in the meantime communication cost in terms of the number of messages and the average size of messages of a scheme should be kept low for efficient execution in a wireless network. Existing multi-party digital signature schemes [1], [3] may not be suitable for low-power devices due to especially high communication cost as power dissipation by an on-board radio transceiver is considered the dominating reason for a short battery life span of these devices.

The first distributed verifiable secret sharing (VSS) is presented in [4], and it is based on Feldman VSS [5] (where each player acts as a dealer). It specifies n parallel runs of all the players, each player selects a random secret $z_i \in \mathrm{GF}(q)$ (a Galois field) and shares it with other players. The players collaboratively construct a non-disqualified set Q in which the secret is shared. The random secret x is set to the sum of the properly received shares from others in Q. In [1], an improved version (in terms of its security) called distributed key generation (DKG) is presented. This protocol can tolerate the attack where an adversary can force the secret key to have a biased distribution in the base field. To do so, an adversary monitors the current disqualified set and response with a complaint to disqualify a particular player such that the last bit of the eventual public key is skewed to 0 with a probability of 3/4 rather than 1/2. This attack is called the GJKR attack (for short) in this paper. Out of total n players, DKG tolerates up to t players under control of a static adversary for $n \geq 2t+1$. However, DKG is expensive and incurs a long latency due to one extra stage right before public key extraction with cost in tantamount to the distributed VSS protocol. In [1], a multi-party signature scheme using the distributed VSS is proposed but with a q_H factor security degradation as compared with that based on keys from DKG, where q_H is the upper bound of queries to the underlying Oracle by an adversary. One challenging problem is to devise a multiparty digital signature scheme with strong security and efficiency for low-power devices.

Existing DKG protocols are based on either the discrete logarithm problem (DLP) over a finite field or the integer factorization problem (IFP). Due to subexponential algorithms to IFP and DLP [6], [7], Elliptic curve cryptosystems (ECC) are safe against common algorithmic techniques. There is no specific subexponential algorithm for elliptic curve discrete logarithm problem (ECDLP) if some precaution is exercised. Shorter keys can be safely used, and with ECC, a small key size means energy savings and latency reduction in wireless communications [8].

In order to enable distributed key generation based on the intractability of ECDLP, the keys are generated from the additive finite group where secret sharing arithmetic operations cannot be directly performed since a finite field structure is essential

Manuscript received November 18, 2007; revised March 9, 2008; accepted June 22, 2008. The associate editor coordinating the review of this paper and approving it for publication is H.-H. Chen.

C. Tang is with Pathfinder Energy Services, Houston, TX 77041 (e-mail: caimu.tang@pathfinderlwd.com).

D. O. Wu is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6130, USA (e-mail: wu@ece.ufl.edu).

A. T. Chronopoulos is with the Department of Computer Science, University of Texas, San Antonio, TX 78249, USA (e-mail: atc@cs.utsa.edu). This research of Dr. Chronopoulos was supported, in part, by a grant from the Center for Infrastructure Assurance and Security of The University of Texas at San Antonio and by NSF CCR-0312323.

C. S. Raghavendra is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA (e-mail: raghu@usc.edu).

Digital Object Identifier 10.1109/TWC.2008.071286

for polynomial interpolation. In this paper, we propose a novel approach to distributed key generation, which enjoys the security stemming from the intractablility of ECDLP as well as computation and communication efficiencies. Our approach to the secret share generation is based on a Galois field so that secret sharing arithmetic operations are still performed in the base Galois field; in contrast, the public information including the public key is expressed as points in the elliptic curve additive group.

It is challenging to design distributed key generation schemes for low-power devices when an adaptive adversary is present, and computing resources are seriously limited on these devices. An adaptive adversary as defined in [9], [10] can select a player to attack based on the adversary's dynamically gathered information at run-time. In each round, the adversary can perform necessary computation and generate messages and send them out before any honest player can. We adopt this type of adaptive model. The difference between an adaptive adversary and a static adversary lies in the decision time when honest players are selected to be attacked. DKG cannot prevent an adaptive adversary from corrupting a player before the public key is extracted [1], [9].

On one hand, in order to add a player to the share group, share refreshing is required so that the added player can contribute to the shares while keeping the public key intact. On the other hand, to protect long-lasting applications, shares are vulnerable to cumulative cryptanalysis attack and partial information leakage. To protect the secret while allowing the secret sharing, proactive share refreshing is one such promising solution (cf. [9], [11] for DLP based systems and [12] for ECDLP based system). There are a few properties to be met for any share refreshing scheme to be useful: (1) the refreshing process does not leak any extra information than the original scheme, (2) the public key has to remain intact before and after share refreshing, (3) the refreshing cost (in terms of number of messages, message sizes and processing overhead) should be a small fraction of the key generation cost. Property (2) will ensure that there is no interruption of operation since users can still use the same public key. Property (3) will make sure that the refreshing process is more efficient than generating a new public key. To the best of our knowledge, this share refreshing issue in the context of distributed key generation has not been addressed.

The main contributions of this paper are: 1) a secret sharing protocol using elliptic curve cryptosystems, 2) a proactive share refreshing protocol, and 3) a multi-party digital signature scheme using these secret sharing and share refreshing protocols. Both the secret sharing protocol and share refreshing protocol are efficient in communications and processing, and the derived multi-party signature scheme is suitable for low-power devices in wireless networks. The secret sharing protocol and the share refreshing protocol are invulnerable to both static and adaptive adversaries, and possess the aforementioned three properties. Our secret sharing protocol uses bit commitment [13] which allows each player to contribute its share in an asynchronous manner. The bit commitment approach significantly reduces the computation overhead as well as the communication cost as compared to DKG. Since our signature scheme is based on keys from a strong key generation protocol, there is no security degradation associated with the proposed scheme. The key share refreshing process finishes in a single round. Hence the signing process of the proposed multi-party signature scheme incurs low latency.

TABLE I
NOTATIONS

ECDLP	: Elliptic curve discrete logarithm problem
IFP	: Integer factoring problem
DKG	: Distributed key generation
A-DKG	: Adaptive DKG [9]
VSS	: Verifiable secret sharing
RTT	: Round-trip time of messages
A-ECDKG	: Adaptive elliptic curve DKG
ECSVP	: Elliptic curve share verification protocol
GJKR	: An attack to VSS from Gennaro et al. [1]
SRP	: Secret share refreshing protocol
MiDS	: Proposed multi-party digital signature
MiDS-S	: Signature signing algorithm of MiDS
MiDS-V	: Signature verification algorithm of MiDS
q	: Order of a non-smooth prime
GF$^*(q)$	: Induced multiplicative group of GF(q)
w_i	: i-th player's identification
Q_i	: Local non-disqualified set by player i
$\mathcal{S}_i$	: $\{w_1, w_2, \cdots, w_{i-1}, w_{i+1}, \cdots, w_n\}$
$E/$GF(q)	: Additive group on a proper curve E
T	: The base point in $E/$GF(q)
p	: Key order
$< T >$	: A group derived from elliptic curve point T
$\{T\}_x$	: Abscissa coordinate of elliptic curve point T
$\mathcal{G}$	: Main subgroup of $< T >$ for cryptography
$\uplus$	: Point add operator over $\mathcal{G}$
$\sum^{\uplus}$	: Point summation under $\uplus$
$f_i(.)$	: Order t polynomial in GF(q) generated by player i
P_{ji}	: A public point generated by player j on $f_i(.)$
s_{ij}	: $f_i(w_j) \in$ GF(q) computed by player i
M_{max}	: Maximum message size

Some of the notations and acronyms that are used throughout this paper are listed in Table I. The rest of paper is organized as follows. The proposed secret sharing and share refreshing protocols are described in Section II. The multi-party digital signature scheme is presented in Section III. Performance analysis of the scheme is presented in Section IV. Section V concludes this paper.

II. Proposed Secret Sharing and Share Refreshing Schemes

We assume that each player has a unique random identification number w_i in GF(q) and players know these numbers of each other. We use w_i for the i-th player or its identification whenever the context is clear. In addition, a point scalar multiplication (a point multiplied by a scalar in GF(q)) and point addition are performed in the elliptic curve main subgroup denoted by $\mathcal{G}$ with a base point T, all other arithmetic operations are performed in finite field GF(q) unless otherwise specified. We also assume that there is another point T' in $\mathcal{G}$ whose discrete logarithm with respect to T is unknown to any of these n players, and for the share refreshing purpose, the duration of the protocol execution is split into equal time slots.

We next describe the adaptive elliptic curve distributed key generation scheme (A-ECDKG). A-ECDKG uses six basic modules, that is, 1) the elliptic curve share verification protocol (ECSVP) on secret shares of non-disqualified players, 2) the key distribution algorithm (KD), 3) the key verification

algorithm (KV), 4) the key check algorithm (KC), 5) the key nonce algorithm (KN), and 6) the key generation algorithm (KG).

Let Q be a set of non-disqualified members, initially set to include all players, and all players in Q precompute a share (d_i for w_i) of a common known number d. The A-ECDKG protocol (in the view of an individual player w_i) is given below:

Protocol 1: A-ECDKG($w_i, Q = \{w_1, w_2, \cdots, w_n\}$)
Execute: set $Q_i = Q$
KD(w_i, t, T)
KV(w_i, t, T)
Erase s_{ji} for $j \in \mathcal{S}_i$ and send d_i
while(1) {
  receive shares of d
  if (at least $t+1$ shares of d are available) {
    compute d using Lagrangian interpolation
    run ECSVP(d, w_i)
  } /* end of if */
  if KN(w_i, t, Q_i) returns false, then run KG(w_i, Q_i) and exit
  } /* end of while */
  KC(w_i, t, Q_i)
} ■

The ECSVP protocol uses the Schnorr three-way proof technique [14] and it follows Canetti's share verification approach in [9] in a quite straightforward manner, and its security is built upon the intractability of ECDLP. ECSVP uses a common challenge d, which is precomputed using DKG in the base finite field GF(q), and at least $t+1$ shares of d are needed during the interpolation process. This common challenge d is then used in ECSVP to verify the shares of players in the non-disqualified set. ECSVP is shown in Protocol 2.

Protocol 2: ECSVP(d, w_i)

(1) Precompute: w_i preselects a one-time random number r_i and broadcasts point $T_i = r_i T$.
(2) Execute:
  (2-a) broadcast
$$u_i = r_i + d(f_i(0)) \tag{1}$$
  (2-b) check whether
$$U_j = T_j \uplus (d(P_{j0})) \tag{2}$$
  where $\uplus$ is the point addition and $U_j = q_i T$. If the equality in (2) does not hold, w_i complains against player w_j.
  (2-c) On receiving a complaint against w_j, broadcast s_{ij}.
(3) If more than t total complaints against w_j are received, w_j is excluded from the non-disqualified set. ■

Remark: The common challenge d in ECSVP cannot be used more than once, the reason is as follows: an adversary can solve (1) for r_i and broadcast adjusted T_i (as a point) in Step (1). Then instead of broadcasting q_i, the adversary broadcasts r_i in Setup (2-a) which would still satisfy (2). However, the above method cannot be employed by an adversary when d is used for the first time since the common d is only available to a player when at least $t+1$ shares of d is made available by other players. Also since all players are literally synchronized when d is made available (all shares of d are broadcast), adversary players colluding on d cannot employ this method either.

Next, we present algorithms of KD, KV, KC, KN, and KG.

Algorithm 1: **KD**(i, t, T) **(at** p_i**)**

1) Initialization: pick $(2t+2)$ random numbers uniformly, $a_{ik} \in \text{GF}(q)$ and $b_{ik} \in \text{GF}(q)$ $(0 \le k \le t)$, as polynomial coefficients to generate two polynomials of degree t as follows:
$$f_i(z) = \sum_{k=0}^{t} a_{ik} z^k, \qquad f_i'(z) = \sum_{k=0}^{t} b_{ik} z^k$$
   - compute $s_{ij} = f_i(w_j) \bmod p$, and $s'_{ij} = f'_i(w_j) \bmod p$ $(j \in \mathcal{S}_i)$.
   - compute $(t+1)$ public values: $P_{ik} = (a_{ik}T) \uplus (b_{ik}T')$ $(0 \le k \le t)$.
2) Dissemination of private information: send a message containing s_{ij} and s'_{ij} to w_j using the private channel between w_i and w_j $(j \in \mathcal{S}_i)$.
3) Dissemination of public information: broadcast a message containing $\{P_{ik} | 0 \le k \le t\}$. ■

Algorithm 2: **KV**(i, t, T) **(at** p_i**)**

Receive s_{ji} and s'_{ji} sent by w_j $(j \in \mathcal{S}_i)$, then for $j \in \mathcal{S}_i$, do the following:

1) verify
$$(s_{ji}T) \uplus (s'_{ji}T') = \sum_{k=0}^{t}{}^{\uplus} \left({w_i}^k P_{jk}\right). \tag{3}$$
2) broadcast a complaint against w_j, if (3) fails for w_j.
3) broadcast s_{ij} and s'_{ij} that satisfy (3), if w_i receives a complaint to him from w_j. ■

Algorithm 3: **KC**(i, t, Q_i) **(at** p_i**)** on locally non-disqualified set Q_i

Update share s_i: w_j is removed from Q_i and update $s_i = \sum_{j \in Q_i} s_{ji}$,
if one of the following two conditions holds:

1) received $t+1$ or more distinct complaints against w_j.
2) received a re-broadcast s_{ji} and s'_{ji}, but the received s_{ji} and s'_{ji} still falsifies (3). ■

Algorithm 4: **KN**(i, t, Q_i) **(at** p_i**)**

1) if $|Q_i| \le t$, return false.
2) broadcast a nonce containing w_i and freeze Q_i.
3) receive nonces.
4) if nonces from all players in Q_i are received, then return true. ■

Algorithm 5: **KG**(i, Q_i) **(at** p_i**)**

Generate public key:

1) compute $A_{i0} = a_{i0}T$ and broadcast A_{i0}.
2) receive A_{j0} $(j \in Q_i)$ and compute public key as $y_i = \sum_{j \in Q_i}^{\uplus} A_{j0}$. ■

Remark 1: Since the process of generating the common challenge d needs to wait for at least $t+1$ DKG shares to be available, the process of A-ECDKG can be considered in functionality as a two-round protocol, where the first round is to obtain the identifiers of the non-disqualified set of players and to wait for shares of d, and the second round is to refine the non-disqualified set by excluding players who fail the

share verification process ECSVP. The criterion to exclude a player is the same as that in DKG in which at least $t+1$ complaints including complaints received before ECSVP are received against an excluded player.

Remark 2: As an alternative, a more efficient method, in terms of communication (eliminating the KN stage), is to use a hard timeout right before the extraction of public key. It is quite straightforward to see that this method is also invulnerable to the GJKR attack. The timeout interval can safely set to the half of the longest round-trip time (RTT) of all pairs of these players in question.

Remark 3: The precomputation involving execution of an instance of DKG is purely for the purpose of multi-party zero-knowledge proof. As an alternative, for a smart-card implementation, this precomputation step can be avoided by simply hardcoding a seed d_i initially to player i's card, and deriving successive d_i from the seed by following a hash chain if the scheme needs to start over after a public key invalidation.

Property 1: A-ECDKG is invulnerable to the GJKR attack.

Proof: By the simulation done in [1], we know that the oracle would have been able to construct the public key without a biased distribution in the base field before the execution of KG when the KN algorithm collects at least $t+1$ nonces and the cardinality of Q_i is greater than t.

Since the honest players will follow the protocol, and their respective non-disqualified set is settled before the execution of KG algorithm. Due to the fact that nonces are already received in all after the public key shares $\{A_{i0}\}$ $(i \in Q)$ are broadcasted at each player. Therefore, the non-disqualified set after the KG algorithm should remain the same since the only messages allowed in the KG algorithm are the (public) values of public key shares. These messages are not able to change the internal state of the protocol, i.e., removal of an honest player from the non-disqualified set. The property follows. ■

We next propose the Share Refreshing Protocol (SRP) as shown in Protocol 3. This algorithm will be executed when a new player joins. The protocol uses four algorithms, namely, the Key Refreshing Distribution algorithm (KRD), the Key Refreshing Verification algorithm (KRV), the Key Refreshing Check algorithm (KRC) and the Key Refreshing Nonce algorithm (KRN).

Protocol 3: SRP: Share Refreshing Protocol at w_i

Execute: set $\hat{Q}_i = Q$
KRD(w_i, t, T)
KRV(w_i, t, T)
while(1) {
  KRC(w_i, t, Q_i)
  if KRN(w_i, t, Q_i) returns false, exit
    if $\mid Q_i \mid < t+1$
      A-ECDKG(w_i, Q)
} ■

Algorithm 6: **KRD**(i, t, T)

1) Initialization: pick t random numbers uniformly, $\hat{a}_{ik} \in \mathrm{GF}(q)$ $(1 \le k \le t)$, as polynomial coefficients to generate a polynomial of degree t as follows:

$$\hat{f}_i(z) = a_{i0} + \hat{a}_{i0} + \sum_{k=1}^{t} (a_{ik} + \hat{a}_{ik}) z^k$$

where, $\hat{a}_{i0}$ is 0 if no new member joins, otherwise, it is just another random number in $\mathrm{GF}(q)$.

- compute $\hat{s}_{ij} = \hat{f}_i(w_j) \bmod p$ $(j(\neq i) \in Q)$.
- compute t public values: $\hat{A}_{ik} = ((a_{ik} + \hat{a}_{ik})T)$ $(1 \le k \le t)$.

2) Dissemination of private information: send a message containing $\hat{s}_{ij}$ to w_j using the private channel between w_i and w_j $(j \in Q$ and $j \neq i)$. For a new member joining, send $\hat{a}_{i0}$ to the new member in the private channel. The new member has the following polynomial:

$$\hat{f}_u(z) = \left(-\sum_{i \in Q} \hat{a}_{i0} \right) + \sum_{k=1}^{t} (\hat{a}_{uk}) z^k$$

where, u is the new member identification index. Further, $A_{u0} = (-\sum_{i \in Q} \hat{a}_{i0})T$ and $\hat{A}_{ik} = (\hat{a}_{uk})T$, and this new member is treated as a regular member of Q herein, i.e., $Q = Q \cup \{w_u\}$.

3) Dissemination of public information: broadcast a message containing $\{A_{i0}, \hat{A}_{ik} | 1 \le k \le t\}$. ■

Algorithm 7: **KRV**(i, t, T)

receive $\hat{s}_{ji}$ sent by w_j $(j \in Q)$, then for $j \in Q$, do the following:

1) verify

$$\hat{s}_{ji} T = A_{j0} + \sum_{k=1}^{t} {}^{\uplus} \left({w_i}^k \hat{A}_{jk} \right). \tag{4}$$

2) broadcast a complaint against w_j, if (4) fails for w_j.
3) broadcast $\hat{s}_{ij}$ that satisfies (4), if w_i receives a complaint to him from w_j. ■

Algorithm 8: **KRC**$(i, t, \hat{Q}_i)$

update share $\hat{s}_i$: w_j is removed from $\hat{Q}$ and update $\hat{s}_i = \sum_{j \in \hat{Q}_i} \hat{s}_{ji}$,
if one of the following two conditions holds:

1) received $t+1$ or more distinct complaints against w_j.
2) received a broadcast $\hat{s}_{ji}$, but the received $\hat{s}_{ji}$ still falsifies (4). ■

Algorithm 9: **KRN**(w_i, t, Q_i)

1) if $|Q_i| \le t$, return false.
2) broadcast a nonce containing w_i and freeze Q_i.
3) receive nonces.
4) if nonces from all players in Q_i are received, then return true. ■

Remark 1: When $\hat{Q} \neq Q$, y is invalidated by SRP, replay of old A_{i0} is not possible since $\hat{A}_{i0}$ $(\neq A_{i0})$ is used in SRP. The security of SRP is provided by the Pedersen VSS since the public key was unbiasedly selected by A-ECDKG.

Remark 2: $\{s_{ji} | j \in Q\}$ at w_i can be removed after they are used in A-ECDKG since SRP does not require it.

Remark 3: In practice, when a new member joins, only $t+1$ players need to readjust their polynomial constants a_{i0} since the protocol can work against up to t colluding players. In SRP, we readjust all players in Q for the sake of simplicity.

Remark 4: In most cases, a public key is generated, a timestamp and expiration date is associated with it. Therefore, before the key is about to expire, a SRP execution is deemed necessary.

Note that up to t corrupted players can maliciously impersonate up to t honest players, thus, when there are at least $t+1$ honest players left in the worst-case, the protocol will not invalidate the public key. Hence the uniqueness of the non-disqualified set of SRP follows.

Denote the common non-disqualified set by $\hat{Q}$, the following Property 2 gives the consistency of the public key of SRP, and Property 3 gives the secrecy of SRP on the shared secret of the group and secret shares of honest players.

Property 2: The public key y is still valid unless the cardinality of $\hat{Q}$ is fewer than $t+1$.

Proof: Consider the following polynomial (5):

$$F(z) = \left(\sum_{k\in\hat{Q}} a_{k0} + \hat{a}_{k0}\right) + \sum_{i=0}^{t}\left(\sum_{k\in\hat{Q}}(a_{ki}+\hat{a}_{ki})\right) z^i \quad (5)$$

where, the coefficients are unknown. The shared secret is the constant term of $F(z)$, i.e., $F(0)$. The public key is $y = F(0)T$. When some player (w_i) in Q is excluded during SRP (i.e., w_i failed the share verification process), its contribution to $F(z)$ is public knowledge since s_{ji} for $j(\neq i) \in Q$ are broadcast. This process is equivalent to the addition of the secret $F(0) - f_i(0)$ by a known constant $f_i(0)$. Since there are at least $t+1$ players, the public key y can be safely used by the non-disqualified players. ■

Property 3: SRP does not leak information of players in the non-disqualified set.

Proof: There are two cases: (1) no member joins and (2) a new member joins. For Case (1), this is the same procedure as Pedersen VSS under the assumption that there are no more than t corrupted players during the refreshing interval since all polynomials are refreshed, i.e., all coefficients of polynomials are random in GF(q). For Case (2), since each member contributes a random piece to the secret a_{u0} and more than t existing players contribute to the new member's shared secret, this does not give away any information with regard to the new member's shared secret (this scheme is $t+1$ sharing). The rest of the argument follows from the simulation of the standard Pedersen VSS. ■

III. Multi-party Digital Signature Scheme Using A-ECDKG

In this section, we show how to sign documents using this A-ECDKG presented in Section II. The approach is to allow a group of individual players (each) generate a partial signature. Then the final signature is combined from these partial signatures. It is named as Multi-party Digital signature (MiDS) scheme. We shall show that fewer than $t+1$ players cannot forge a signature. MiDS is the elliptic curve version of the Schnorr signature scheme [14] and the corresponding threshold Schnorr signature scheme [1] under A-ECDKG. It is open to build a threshold signature scheme other than Schnorr type using this distributed key generation technique, e.g., El-Gammal type.

Denote the message to be signed by m, and the partial signature from w_i by sig_i and the final complete signature by sig. Algorithm 10 and Algorithm 11 show that $t+1$ players can sign a document which can only be verified using the group public key y generated by A-ECDKG. We denote MiDS-S and MiDS-V as the signing algorithm and verifying algorithm of MiDS, respectively. We assume that there are n players and w_i has a secret share s_i and the public key y is certified and is known to all. By the protocol of A-ECDKG, the public share y_i of w_i can be computed using only public information and it equals to s_iT ($y_i = s_iT$). $H(.,.)$ in Algorithm 10 is the collision-resistant one-way hash function which takes two arguments of the same bit length as input.

Algorithm 10: **MiDS-S**(w_i, m, s_i, y)

1) run A-ECDKG to generate a one-time secret share k_i, the corresponding one-time public key $r = kT$, and the one-time public share $r_i = k_iT$. (Note that k is unknown to any player by the properties of A-ECDKG.)
2) compute $c = H(m, \{r\}_x)$, where $\{r\}_x$ is for x-coordinate of point r.
3) broadcast the following sig_i,
$$\text{sig}_i = k_i - c\prod_{l\neq i}\left(w_l(w_i - w_l)^{-1}\right) s_i. \quad (6)$$
where, $(w_i - w_l)^{-1}$ is the precomputed multiplicative inverse element of $(w_i - w_l)$ in GF$^*(q)$.
4) receive partial signatures: if the following received partial signature set SIG_i has cardinality fewer than t, return failure.
$$\text{SIG}_i = \left\{ \text{sig}_j \,\middle|\, (\text{sig}_j)T = r_j - c\sum_{l\neq j} w_l(w_j - w_l)^{-1} y_j \right\}$$
5) generate signature: pick t partial signatures from SIG_i and add them to sig_i to generate the complete signature (c, sig). ■

Algorithm 11: **MiDS-V**$(c, \mathbf{sig}, m)$

1) compute $r = ((\text{sig})\ T) \uplus ((-c)y)$.
2) verify $c = H(m, \{r\}_x)$, accept the signature if it passes the test. ■

Note that one of the reasons to use a random one-time secret upon which all player agree, via A-ECDKG, is to ensure that a nontrivial one-time secret is used and no information regarding the secret shares can be revealed. SRP can be used to refresh the secret shares periodically or allow a new signee to join an existing group, and A-ECDKG enables secret sharing in the presence of static and adaptive adversaries. Hence MiDS is flexible and adaptively secure; moreover, MiDS is efficient since secret shares can be generated in A-ECDKG with much less communication cost in terms of data exchanges and the number of messages required as compared to DKG [1].

The verification of message m for given signature (c, sig) simply follows the verification of Schnorr signature scheme as the actual steps are shown in Algorithm 11. Notice that for any $(t+1)$ shares, we have (which cannot be computed directly in the protocol by any single signee)

$$\text{private key} = \sum_{i=1}^{t+1}\left(\prod_{j=1,j\neq i}^{t+1}\left(w_j(w_i - w_j)^{-1}\right) s_i\right) \mod p,$$

the correctness of the verification process then follows using the same argument as in [14].

Due to the intractability of ECDLP and the elliptic curve Diffie-Hellman problem (i.e. compute rsT from point sT

and rT), the signature has the non-forgery property against any adversary. Property 4 shows the threshold non-forgery property of MiDS.

Property 4: The non-forgeability of MiDS: fewer than $t+1$ players cannot forge the signature.

Proof: We first show that a reduction from MiDS to the Schnorr scheme is possible in polynomial time with a finite number of messages. Then the non-forgery property under the chosen message attack follows from that of Schnorr scheme in which the existential forgery can possibly lead to a non-negligible probability of success of finding the discrete logarithm of ECDLP in polynomial time.

By the secrecy of A-ECDK and Property 1, k is unknown to all players and r follows a uniform distribution in $\mathcal{G}$. Therefore c follows a uniform distribution in the base field. Due to the lack of the required number of degree of freedoms, simultaneously solving systems of equations in the form of (6) is not possible even with two additional constraints: $k = k_1 + k_2 + \cdots + k_n$, and the Lagrangian interpolation equation (i.e., $t+1$ secret shares when combined using Lagrangian interpolation are equal to the private key s).

Without loss of generality, when the first t players collude, to generate a valid signature, we need to obtain the unknowns k_{t+1} and s_{t+1} which are subject to (6). Due to the fact that k is unknown, and the shared secret is unknown, this system still lacks one degree of freedom. Since the simulator cannot access more than t players during one course of execution, it is not able to directly generate the valid sig for a given c which can pass the test in Algorithm 11.

With the random oracle, the simulator can initiate $t+1$ parallel runs with randomly selected $t+1$ signees. For each run, the simulator randomly feeds messages to Algorithm 10 to obtain two partial signatures $\{\text{sig}'_i, \text{sig}''_i\}$. When these $\{\text{sig}'_i\}$ and $\{\text{sig}''_i\}$ $(1 \le i \le t+1)$ are combined as in Algorithm 10, it can yield two instances with two valid signatures (c', sig') and (c'', sig''). Let

$$r = ((\text{sig}')T) \uplus ((-c')y) = ((\text{sig}'')T) \uplus ((-c'')y). \quad (7)$$

From Lemma 2 in [15], we know this probability is non-negligible. From (7), following the same argument in the proof of Theorem 3 in [15] in the base field GF(q) instead of $Z/_pZ$, the ECDLP can be solved with non-negligible probability of success. The solution to $\log(y)$ is then $\log(y) = (\text{sig}' - \text{sig}'')(c' - c'')^{-1}$, Therefore, the intractability of ECDLP ensures the non-forgeability of MiDS. ∎

IV. Performance Analysis

Since Koblitz curves are used in A-ECDKG, SRP and MiDS for our quantitative results, a brief description of Koblitz curve is given as follows. This type of curves has the following form: $y^2 + xy = x^3 + ax^2 + 1$, where $a \in$ GF(2). The domain parameters are given as follows: 1) Threshold value t and one field element for the coefficient a corresponding to a unique quintuple. 2) Field representation type, for polynomial basis, the irreducible polynomial and its coefficients are required; for normal basis, which is most efficient for raising the unique identification number w_i to a power fewer than $(t+1)$, the base element θ of the basis is needed. 3) Point representation, point compression type, coordinate system type and the selected point whose order must have a large prime factor p ($p > 2^{160}$) which is almost guaranteed for an ABC curve. 4) Cofactor h which can be either 2 or 4 since m has to be a prime, and h is multiplied by p which results in the total number of points in the group $E/\text{GF}(2^m)$.

TABLE II
Computation Cost

Alg.	MOD Multiply	MOD Add	Compare
KD	$(2(t+1), 2\omega_1)$	$(t, 2nt)$	(0, 0)
KV	$((n-1)(t+3), \omega_1)$	$(n-1)(t+1), 0)$	$(n-1, 0)$
KC	$(n(t+3)/3, 0)$	$(n(t+1)/3, n-1)$	(0, 0)
KG	$(1, 0)$	$(n-1, 0)$	(0, 0)
KN	$(0, 0)$	$(0, 0)$	(0, 0)
KRD	$(t+1, \omega)$	$(0, 2nt+t$	(0, 0)
KRV	$((n+1)t, 0)$	$(nt, 0)$	$(n-1, 0)$
KRC	$((n+1)t/3, 0)$	$(0, n+1)$	(0, 0)
KRN	$(0, 0)$	$(0, 0)$	(0, 0)
ECSVP	$(2n-2, 1)$	$(n, 1)$	$(n-1, 0)$
MiDS-S	$(2t+1, 1)$	(t, t)	(t, 0)
MiDS-V	$(2, 0)$	$(1, 0)$	(0, 1)

TABLE III
Worst-Case Communication Cost

Algorithm	Rx	Tx	M_{max}
KD	0	$(n-1, 1)$	$(2\log_2(q), \omega')$
KV	$(n+t+1)$	$(0, 2t)$	$(0, 2\log_2(q))$
KC	$t(t+2)$	$(0, 0)$	$(0, 2\log_2(q))$
KG	$n-1$	$(0, 1)$	$(0, \log_2(p))$
KN	$n-1$	$(0, 1)$	$(0, \log_2(q))$
KRD	0	$(n, 1)$	$(2\log_2(q), \omega')$
KRV	$n+1$	$(0, 2t)$	$(0, 2\log_2(q))$
KRC	$t(t+2)$	$(0, 0)$	$(0, 2\log_2(q))$
KRN	n	$(0, 1)$	$(0, \log_2(q))$
ECSVP	$n+t$	$(0, 2)$	$(0, \log_2(q))$
MiDS-S	t	$(0, 1)$	$(0, \log_2(q))$
MiDS-V	0	$(0, 0)$	$(0, 0)$

We next analyze the overhead of A-ECDKG, SRP, and MiDS including computation, communications and memory. Table II shows the computation cost excluding precomputation cost, where $\omega_1 = (n-1)(t+1)\log_2(t+1)$, $\omega = n(t+1)\log_2(t+1)$. Each cell in Table II is in (x, y) format, where x is the measure for the field arithmetic cost on $\mathcal{G}$ – the elliptic curve main subgroup, and y is the corresponding arithmetic cost on GF(q). Note that the hash cost is excluded for MiDS-S and MiDS-V in Table II. Our bounds are not very tight for large t, and for large t, Stirling approximation on the bounds to the cost of polynomial evaluations using repeat squaring could lead to tighter upper bounds. For those refreshing algorithms, we use the case where one member joins an existing group with a total number of $(n+1)$ players including the new member. Table III shows the number of messages involved in these algorithms and protocols where $\omega' = (t+1)\log_2(p)$. The transmission column and message size column in Table III consist of two expressions in (x, y) format, where x is the transmission cost or sizes of messages via the private channel, y is the corresponding measure via the broadcast channel. Note that the reception message sizes can be derived by adding corresponding private message sizes with public message sizes.

Note that in Table II, for the computation costs of MiDS-S and MiDS-V, the cost of key generation is excluded. In

TABLE IV
COST COMPARISON ON COMMUNICATIONS

Scheme	Rx	Tx	M_{max}
A-ECDKG	$4n + t^2 + 4t - 1$	$(n-1, 2t+5)$	$\Psi(q,p)$
DKG	$4n + 2t^2 + 6t$	$(n-1, 4t+4)$	$\Psi(q',q')$
SRP	$2n + t^2 + 2t$	$(n, 2t+2)$	$\Psi(q,p)$
A-DKG	$5n + 2t^2 + 7t$	$(n-1, 4t+6)$	$\Psi(q',q')$

TABLE V
COST COMPARISON ON SIGNATURE SCHEMES

Scheme	Reception	Transmission	M_{max}
MiDS	t	1	$\log_2(q)$
DKG Based	t	1	$\log_2(q')$
Adaptive DKG Based	t	1	$\log_2(q')$
Pedersen VSS Based	t	1	$\log_2(q')$

Table II, the evaluation of an exponent in GF(q) uses repeated squaring and the evaluation of point scalar multiplication in $\mathcal{G}$ uses τ-adic Non-Adjacent Form which eliminates point doubling and converts a point scalar multiplication to point addition operations. Point scalar multiplication can be performed in a fraction of a millisecond in a typical embedded processor (with a moderately capable finite field coprocessor [16]) at 200 MHz for National Institute of Standards and Technology (NIST) anomalous binary curve (ABC) [17] with 163 bit key in GF(2^{163}) (i.e., the main subgroup also has an order of length of 163 bits). Table IV shows the overall worst-case communication cost comparison among A-ECDKG, DKG and adaptive DKG [9], where q' is the key length of DKG, and $\Psi(x,y) = (2\log_2(x), (t+1)\log_2(y))$. Since $\log(q) \sim \log(p) \sim 1/6 \log(q')$ for practical applications with a similar level of security [8], A-ECDKG can conserve more energy when compared to either DKG or adaptive DKG. Similar results hold for MiDS as shown in Table V.

Note that although in Table III the reception overhead in the KC and KRC algorithms are included in a loop, the number of received messages must be bounded above by the total number of messages actually transmitted. In Table III, the message sizes vary. However, round-trip time (RTT) (over Internet via wireless access) is the dominating factor on latency incurred during a successful key generation with a typical hardware implementation with a coprocessor. On energy consumption, the message sizes matter the most on both the radio transmitter and the receiver. Compared to DKG over finite afield, message sizes of public information in KD are reduced by a factor less than 1/6, and message sizes of other messages represented by elements in the base field are also reduced proportionally under the same level of security since a smaller finite base field is used by A-ECDKG than that by DKG. Likewise, the sizes of these partial signature messages generated by MiDS-S are also reduced by a factor less than 1/6 than those in other known schemes [1], [3].

Since the signing process of MiDS only requires a 1/2 of message round-trip time to collect these partial signatures after the secret shares and corresponding public key are generated, message transmission delay dominates the time to generate the signature by each signee. As an example, in a typical embedded processor at 200 MHz with a coprocessor, the partial signature can be obtained in around 51 ms on 163 bit NIST ABC curve with a 100 ms RTT after the distributed keys are made available by A-ECDKG, and the actual signature can be generated in 75 ms for $t = 4$ after precomputation which incurs once after an A-ECDKG or a single SRP execution. MiDS-V requires merely two scalar multiplication operations and one addition operation of elliptic curve points. This can be completed in less than 10 ms on the same hardware platform as above.

TABLE VI
WORST-CASE MEMORY REQUIREMENTS

Algorithm	Worst-Case Memory
KD	$(t+1)\log_2(p) + 2(t+1)\log_2(q) + C$
KV	$2(n-1)\log_2(q) + C$
KC	$\log_2(q) + C$
KN	0
KG	$n\log_2(p)$
KRD	$(t+1)\log_2(p) + 3(t+1)\log_2(q) + C$
KRV	$n\log_2(q) + C$
KRC	$\log_2(q) + C$
KRN	0
ECSVP	$\log_2(p) + \log_2(q) + C$
MiDS-S	$2n\log_2(p) + n\log_2(q) + C$
MiDS-V	C

Table VI shows the worst-case memory requirements for each algorithm of A-ECDKG, SRP and MiDS, where C is the buffer size independent of n and t for intermediate results. In the worst case, the memory requirement at each signee linearly depends on the total number of signees during key generation and share refreshing. Due to the fact that hardware configuration is set up for the worst case in common practice, signing and verifying processes do not affect the hardware configuration. In general, for a moderate group size of signees in practice, common hand-held devices can satisfy these memory requirements. Due to the shorter key in A-ECDKG, the worst-case memory requirements of A-ECDKG and MiDS are also shorter than those of DKG and DKG based signature scheme, respectively.

V. CONCLUSION

In this paper, a novel and efficient multi-party signature scheme is proposed. The protocol is based on A-ECDKG which is invulnerable to both static and adaptive adversaries. This protocol is efficient and suitable for hand-held devices in a wireless network.

REFERENCES

[1] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," *J. of Cryptology*, vol. 20, no. 1, pp. 51-83, 2007.

[2] C. Tang, A. T. Chronopoulos, and C. S. Raghavendra, "Soft-timeout distributed key generation for digital signature based on elliptic curve D-log for low-power devices," in *Proc. IEEE SecureComm*, IEEE Press, 2005, pp. 353-264.

[3] J. van der Merwe, D. S. Dawoud, and S. McDonald, "A fully distributed proactively secure threshold-multisignature scheme," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 4, pp. 562-575, 2007.

[4] T. Pedersen, "A threshold cryptosystem without a trusted party," in *Proc. Eurocrypt '91*, Springer-Verlag, 1991, pp. 522-526.

[5] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *Proc. 28th IEEE FOCS*, IEEE Press, 1987, pp. 427-437.

[6] C. Pomerance, "Analysis and comparison of some integer factoring algorithms," in *Computational Methods in Number Theory*, Mathematisch Centrum, 1982, pp. 89-139.

[7] A. M. Odlyzko, "Discrete logarithm in finite fields and their cryptographic significance," in *Proc. Eurocrypt '84*, Springer-Verlag, 1985, pp. 224-314.
[8] K. Lauter, "The advantages of elliptic curve cryptography for wireless security," *IEEE Wireless Commun. Mag.*, vol. 11, no. 1, pp. 62-67, 2004.
[9] R. Canetti, R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Adaptive security for threshold cryptosystems," in *Proc. Crypto'99*, Springer-Verlag, 1999, pp. 98-115.
[10] M. Abe and S. Fehr, "Adaptively secure Feldman VSS and applications to universally-composable threshold cryptography," in *Proc. CRYPTO'04*, Springer-Verlag, 2004, pp. 317-334.
[11] A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive public key and signature systems," in *Proc. ACM CCS*, ACM, 1997, pp. 100-110.
[12] C. Tang and D. O. Wu, "An efficient proactive share refreshing scheme for secret sharing in distributed systems," in *Proc. IEEE Globecom*, IEEE Press, 2006.
[13] M. Blum, "Coin flipping by telephone," in *Proc. 24th IEEE Compcon*, Springer-Verlag, 1982, pp. 133-137.
[14] C. P. Schnorr, "Efficient signature generation by smart cards," *J. of Cryptology*, vol. 4, no. 3, pp. 161-174, 1991.
[15] D. Pointcheval and J. Stern, "Security proofs for signature schemes," in *Proc. Eurocrypt '96*, Springer, 1996, pp. 387-398.
[16] M. Ernst, M. Jung, F. Madlener, S. Huss, and R. Blumel, "A reconfigurable system on chip implementation for elliptic curve cryptography over GF(2^n)," in *LNCS, vol. 2523*, Springer-Verlag, 2002, pp. 381-399.
[17] (2001) ANS X 9.63-2001 "Public key cryptography for the financial services industry: key agreement and key transport protocols using elliptic curve cryptography," American National Standards Institute.

Caimu Tang (S'97–M'05) received B.S. in Applied Mathematics from Xi'an Jiaotong University (within three years and the 1st rank in class), Xi'an, China, in 1990, M.S. in Computer Science from Wayne State University, Detroit, Michigan, in 1997, and Ph.D. in Computer Science from University of Southern California, in 2005. From Aug. 2005 to April 2006, He was with Rockwell Scientific Company at Thousand Oaks, CA, as a research scientist. From Aug. 2006 to June 2007, he was with Tut Systems, Lake Oswego, OR, as a staff engineer. Since June 2007, he has been with Pathfinder Energy Services, Houston, TX, as a DSP engineer. His research interests are in the areas of source coding, signal processing for real-time measurement/logging-while-drilling, network security, video coding/transcoding, and wireless communications.

Dapeng Oliver Wu (S'98–M'04–SM'06) received B.E. in Electrical Engineering from Huazhong University of Science and Technology, Wuhan, China, in 1990, M.E. in Electrical Engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1997, and Ph.D. in Electrical and Computer Engineering from Carnegie Mellon University, Pittsburgh, PA, in 2003.

Since 2003, he has been on the faculty of Electrical and Computer Engineering Department at University of Florida, Gainesville, FL, where he is currently Associate Professor. His research interests are in the areas of networking, communications, multimedia, signal processing, and information and network security. He received ONR YIP award in 2008, NSF CAREER award in 2007, the IEEE Circuits and Systems for Video Technology (CSVT) Transactions Best Paper Award for Year 2001, and the Best Paper Award in International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine) 2006.

Currently, he serves as the Editor-in-Chief of the JOURNAL OF ADVANCES IN MULTIMEDIA, and an Associate Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and INTERNATIONAL JOURNAL OF AD HOC AND UBIQUITOUS COMPUTING. He was an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY between 2004 and 2007. He is also a guest-editor for IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC), Special Issue on Cross-layer Optimized Wireless Multimedia Communications. He has served as Program Chair for IEEE International Conference on Communications (ICC 2008), Signal Processing for Communications Symposium, and as a member of executive committee and/or technical program committee of over 50 conferences. He is Vice Chair of Mobile and wireless multimedia Interest Group (MobIG), Technical Committee on Multimedia Communications, IEEE Communications Society. He is a member of the Best Paper Award Committee, Technical Committee on Multimedia Communications, IEEE Communications Society.

Anthony T. Chronopoulos (SM'98) received his Ph.D. at the University of Illinois in Urbana-Champaign in 1987. He is a senior member of IEEE. He has published 44 journals and 55 refereed conference proceedings publications in the areas of distributed systems, game theory, networks and security, parallel processing. He has been awarded 15 federal/state government research grants. His work is cited in more than 375 non-co-authors' research articles. He is a professor of Computer Science at the University of Texas at San Antonio, TX, U.S.A.

Cauligi S. Raghavendra (F'97) received the B. Sc. (Hons.) physics degree from Bangalore University, India, in 1973, the B.E. and M.E. degrees in electronics and communication from the Indian Institute of Science, Bangalore, in 1976 and 1978, respectively, and the Ph.D. degree in computer science from the University of California at Los Angeles in 1982. He is a Professor of Electrical Engineering and Computer Science and Senior Associate Dean for Strategic Initiatives of the Viterbi School of Engineering at the University of Southern California, Los Angeles. Dr. Raghavendra was a recipient of the Presidential Young Investigator Award for 1985.