A PROVABLY SECURE GROUP KEY AGREEMENT PROTOCOL USING ELLIPTIC CURVE DIFFIE-HELLMAN

Presentation by Srinivasa Naresh V

September 28, 2015

Overview

- Contributations from research work done
- Motivation
- Methodology and Problem Statement
- Major Contributions
- Unauthenticated CGKA Protocol: Paper-1
- $DACGKA^{PP-PKI}$: Paper-2.
- $DACGKA^{Int-Sgn}$: Paper-3

Contributions from research work done

Details of Publications:

First Paper:Accepted and Published

- "Elliptic curve based dynamic contributory group key agreement protocol for secure group communication over ad hoc networks" is published in *International Journal of Network security.*
- Date of submission: Feb. 6, 2013.
- Date of acceptance: Feb. 6, 2014.
  Journal Details:
- Name of the journal: International Journal of Network security. *Scopus Elsevier* (Web Science) indexing and Free Journal.
- Source Normalized Impact per Paper (SNIP): 1.987 (2013).
- SCImago Journal Rank (SJR): 0.543 (2013).
- Impact Factor for year 2013: 1.3921.
- H-index: 20

Second Paper:Accepted

- "A new two-round dynamic contributory group key agreement protocol using elliptic curve Diffie-Hellman technique" submitted to SADHANA *Springer Verlag with ISI impact factor 0.587*)(The URL is http://sadh.edmgr.com/) on 5thJune, 2013.
- Date of submission : 5th June, 2013
- Date of acceptance : 14th June 2015
- *Journal Details:*
- Name of the journal: SPRINGER VERLAG-SADHANA
- **Science Citation Index and Scopus Elsevier (Web Science) indexing Free Journal**
- Source Normalized Impact per Paper (SNIP): 0.99 (2013)
- SCImago Journal Rank (SJR): 0.26 (2013)
- **Thomson's Reuters Impact Factor for year 2013:** 0.587 (2013)
- H Index: 25

Third Paper: Accepted with minor revisions

- Paper entitled "Provably Secure Group Key Agreement Protocol based on ECDH with integrated signature" is under revision.
- Journal title: Wiley-Secure Communication Networks
- **Science Citation Index and Scopus Elsevier (Web Science) indexing Free Journal.**
- **Thomson Reuters Impact Factor:0.72**
- Date of submission: May. 7th, 2015.
- Date of acceptance: August 7th, 2015 (recommended for publication with a minor revision and the revised paper is submitted)

Best Research Paper and UGC Sanctioned Project Fifth Paper:

- "Elliptic curve cryptography-Diffie Hellman Technique extended for multiple two party keys at a time" was selected as the best research paper among the submitted papers for the presentation in Inter University Faculty Forum and presented on 23.11.2013 through video conference facility located at 'C' block in A.P Secretariat, Hyderabad Organized by the then Government of Andhra Pradesh (APSCHE and CCE).
- UGC Minor Research Project Titled: "A provably secure dynamic authenticated contributory group key agreement Protocol using elliptic curve Diffie-Hellman" was sanctioned with a financial assistance of RS 3,70,000.

Motivation

- **Wireless networks** are growing rapidly in the last few years and secure and reliable communication mechanisms are always active research area with growing popularity in **group oriented and collaborative applications**.
- In the light of the rapid spreading of the group ware applications based on mobile ad-hoc networks (MANETS), the need for Secure Group Communications (SGCs) is growing day by day.
- Providing SGC over MANETS is a very difficult task because they are mostly without much infrastructure.
- **ECC** emerged as the **cryptographic choice** for ad-hoc networks and communication devices because it can provide high security with very smaller key sizes and also at low computational expenses.

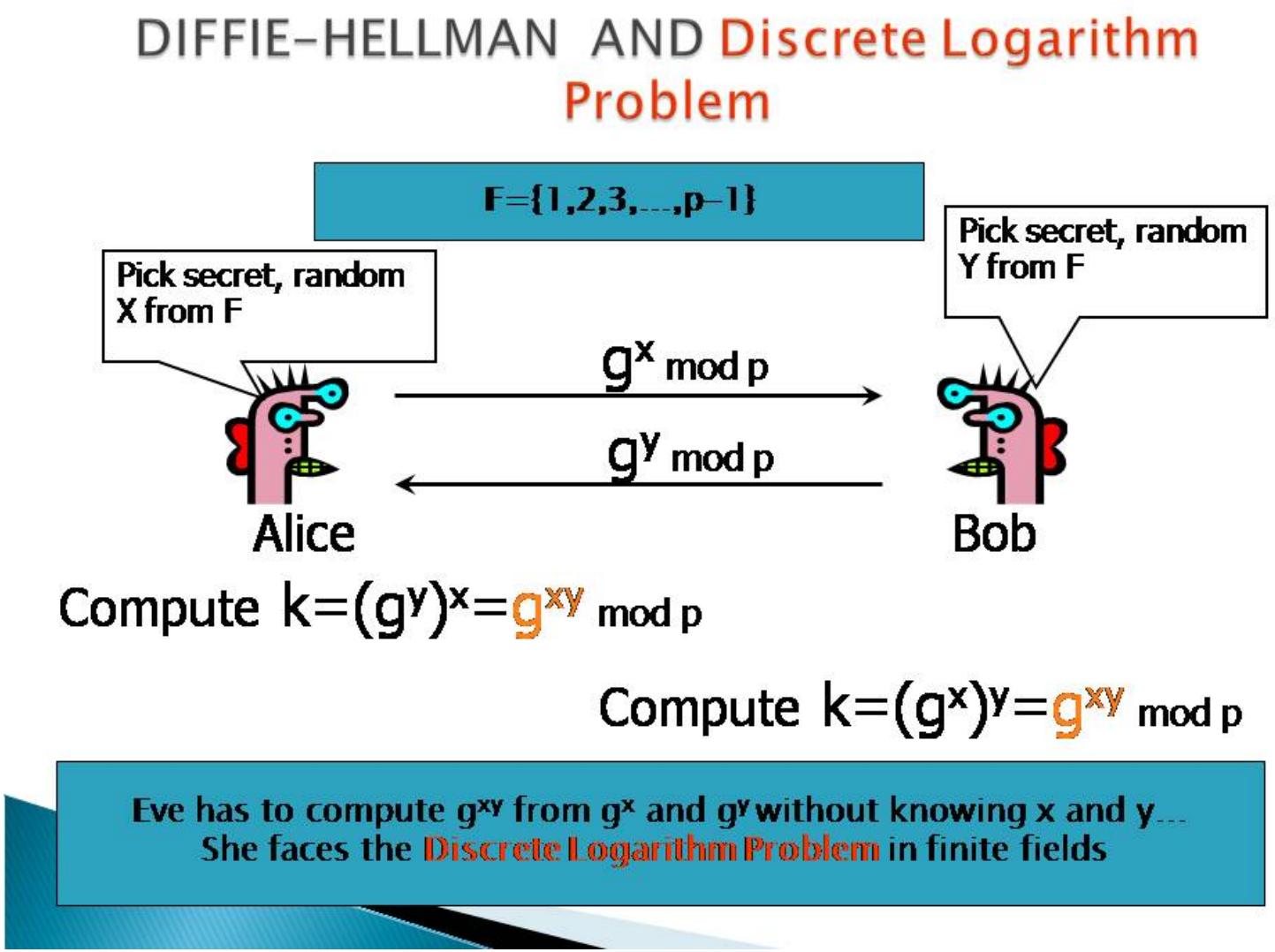
- The **simplicity and elegance of D-H** motivated many researchers to extend D-H to group settings and we are no exception. It is with this spirit that we attempted to extend the same ECDH with integrated signature scheme to group settings with established formal security model and also with most of the desirable security attributes.
- Further, Two-party communication can be viewed as a **discrete phenomenon**: it starts, lasts for a while and ends. OTOH, Group communication is more complicated: the groups start, mutate as members leave and join and there might not be a **well defined end**. This complicates attendant security services, in particular, for key management.

Applications of ECC

Where can we apply ECC?

- Wireless communication devices.
- Mobile ad-hoc networks (MANETS).
- Wireless sensor networks (WSN).
- Smart cards.
- Web servers that need to handle many encryption sessions.
- Any application where security is needed but lacks the power, storage and computational power that is necessary for our current cryptosystems.

DIFFIE-HELLMAN AND Discrete Logarithm Problem



Elliptic Curve Diffie–Hellman Exchange

Alice and Bob want to agree on a shared key.

Alice and Bob compute their public and private keys.

Alice

Private Key = a

Public Key = $P_A = a * B$

Bob

Private Key = b

Public Key = $P_B = b * B$

Alice and Bob send each other their public keys.

Both take the product of their private key and the other user's public key.

Alice → $K_{AB} = a(bB)$

Bob → $K_{AB} = b(aB)$

Shared Secret Key = $K_{AB} = abB$

Elliptic Curve DIFFIE-HELLMAN

Elliptic Curve Discrete Logarithm problem

- **Hard problem**" analogous to discrete log
  - Q=kP, where Q, P belong to a prime curve
    given k, P → "easy" to compute Q
    given Q, P → "hard" to find k
  - known as the elliptic curve logarithm problem
    - k must be large enough
- ECC–DH security relies on elliptic curve logarithm problem
  - compared to factoring, can use much smaller key sizes than with RSA etc

  → for similar security ECC offers significant computational advantages
- The discrete logarithm problem on elliptic curve groups is believed to be more difficult than DLP the corresponding problem in (the multiplicative group of nonzero elements of) the underlying finite field.

Comparative Analysis between ECDLP-based scheme and DLP-based scheme

Table: Key sizes

ECDLP-based scheme (size of n in bits)	DLP-based scheme (modular size in bits)
112	512
160	1024
224	2048
256	3072
384	7680
512	15360

Objectives:

The main objective of the thesis is to propose "A provably secure dynamic authenticated contributory group key agreement protocol using elliptic curve Diffie-Hellman" addressing some of the key issues in GKA protocols such as, to

- Protect, **subsequent sessions** from the leaving members and the **previous sessions** from the joining members.
- Minimize **communication costs** such as the number of rounds, number of messages, etc.
- Minimize **computational costs** such as the number of exponentiations for DLP-based protocols, number of scalar multiplications for ECDLP-based protocols, signatures, and verifications etc.
- Minimize **group key updates** associated with join and leave protocols
- Stand up against both **active and passive attacks**.
- Establish a **formal security model**.

Methodology and Problem Statement:

Methodology

- As sensor networks become one of the key technologies to realize ubiquitous computing, security remains a growing concern. Although a wealth of key-generation methods have been developed during the past few decades, they cannot be directly applied to sensor network environments for want of infrastructure.
- **Elliptic Curve Cryptography (ECC)** has emerged as a suitable public key cryptographic foundation for **constrained environments**, providing strong security for relatively small key sizes.

Problem Statement: With the above objective in mind, in this thesis "A new two round authenticated contributory group key agreement based on Elliptic Curve Diffie-Hellman" protocol for secure group communication over ad-hoc networks is introduced and extended for Dynamic Peer Groups (DPG) with join and leave protocols.

The major contributions:

- The major contribution of this thesis is DACGKA, a **Family of Group key Agreement protocols** for DPGs in a formal security model with optimal communicational and computational costs.
- **Relevance to ad hoc networks:**
  1. This protocol is well suited to ad hoc networks as it requires *no special ordering* of the participants.
  2. For each execution of the protocol, a **random participant** is chosen as the group leader. It is robust as loss of messages from some participants towards the leader, does not prevent other participants from calculating the group key.
  3. It has **efficient join and leave** protocols to handle dynamism in ad hoc networks.
  4. Also the bulk of the computation can be assigned to more powerful devices, as most ad hoc networks are expected to be composed of devices of **unequal computing power.**
- **Simple and Efficient:** The protocol along with the join and leave procedures is not only simple and efficient but also tightest with a proof of security in a **standard model** under the Decisional Diffie-Hellman Assumption.

The major contributions:

- The most difficult task of providing SGC over ad-hoc mobile networks is addressed in this thesis, as the proposed protocol being *ECDLP-based one*, making it less expensive and also more **suitable to resource constraint ad-hoc/sensor networks** especially in view of its smaller key sizes and the recent advances in the computational prowess.

  Proposed Protocols in this thesis:
- Un Authenticated Contributory Group Key Agreement (UA-CGKA) Protocol.
- $DACGKA^{PP-PKI}$: Dynamic Authenticated Contributory Group Key Agreement (DACGKA) Protocol using *ECDH with PP-PK*I.
- $DACGKA^{Int-Sgn}$: Dynamic Authenticated Contributory Group Key Agreement (DACGKA) protocol using *ECDH with integrated signature*.

1.Unauthenticated CGKA Protocol: Paper-1

Round 1:

M_1	M_2	...	M_i	...	M_n
•	•	...	•	...	•
x_1	x_2	...	x_i	...	x_n
$x_1.P$	$x_2.P$	...	$x_i.P$	...	$x_n.P$

Communication: $M_1 \xrightarrow{x_1.P} M_i, 2 \leq i \leq n.$

$$M_i \xrightarrow{x_i.P} M_1, 2 \leq i \leq n.$$

Computation: $M_1 : K_{1i} = x_1 X_i = x_1(x_i.P) = x_1 x_i.P = (x_{K_{1i}}, y_{K_{1i}}), 2 \leq i \leq n.$

$$M_i : K_{1i} = x_i X_1 = x_i(x_1.P) = x_i x_1.P = (x_{K_{1i}}, y_{K_{1i}}), 2 \leq i \leq n.$$

Hence take $x_{K_{1i}}$ as $(m-1)$ shared keys between the GC (M_1) and each of the remaining group member M_i, $2 \leq i \leq n$ respectively.

Unauthenticated CGKA Protocol:

Round 2:

Computation:

$M_1 : L_i = [\prod_{j=2, j\neq i}^{n} x_{K_{1j}}]P, 2 \leq i \leq n.$

Communication: $M_1 \xrightarrow{L_i} M_i, 2 \leq i \leq n.$

Group Key Computation:
M_i: $K = [x_{K_{li}}]L_i = [x_{K_{li}}][\prod_{j=1, j\neq i}^{m} x_{K_{lj}}]P = [\prod_{i=1}^{m} x_{K_{li}}]P = (x_K, y_K).$

M_1: Since the GC knows all the shared keys, it also generates the group key as follows:

$K = [\prod_{i=1}^{m} x_{K_{li}}]P = (x_K, y_K).$

Hence x_K is now a group key among the group members.

Dynamic Contributory Group key Agreement Protocol (DCGKA):

- CGKA addresses group key agreement for static groups. However, often times it becomes necessary to either **add a new member** (or) **remove an existing one** in to the presently communicating group.

- Naturally, it is desirable to do so without executing the entire protocol a new. This issue is addressed in the extension of CGKA to DCGKA by proposing join protocol and leave protocol with proper security mechanisms.

Join protocol

The foremost security requirement in member addition is the concealment of the prior group key with respect to outsider and newly joined group members.

1. As soon as a new member M_{n+1} wants to join the group, intimates the GC and generates a ECDH-style key $x_{K_{l,n+1}}$ with GC.
2. GC generates a random number $R^{'}_{n+1}$ and broadcasts $\left[x_{K_{l,n+1}}.R^{'}_{n+1}\right]$ P to all the previous members of the group M_i. On receiving they compute the new key.

$$NJGK = [x_K.x_{K_{l,n+1}}.R^{'}_{n+1}]P = [(\prod_{i=1}^{n+1} x_{K_{l,i}}.R^{'}_{n+1})]P \quad (1)$$

where x_K is Previous Group Key.

3. GC transmits $[x_K \ R^{'}_{n+1}]P$ to M_{n+1} then M_{n+1} computes the new key as follows

$$NJGK = [x_K.x_{K_{l,n+1}}R^{'}_{n+1}]P = [(\prod_{i=1}^{n+1} x_{K_{l,i}}.R^{'}_{n+1})]P \quad (2)$$

Leave Protocol

The foremost security requirement in member leaving is the concealment of the succeeding (future) group key with respect to both outsiders and former group members.

1. As soon as M_j wants to leave the group, it intimates the GC and then GC, M_1 generate a random number R'_j.
2. M_1 sends $\left[R'_j x_{K_{l,j}}^{-1}\right] P$ by encrypting with $x_{K_{l,i}}$ to the corresponding group member M_i, $i \neq j$,(i.e), except leaving member.

$$M_1 \xrightarrow{E_{K_{l,i}}\left[R'_j x_{K_{l,j}}^{-1}\right] P} M_i, for 2 \leq i \leq n, i \neq j. \tag{3}$$

After receiving, respective members decrypt the message with their respective keys $x_{K_{1,i}}$ and compute the new key as follows.

$$NLGK = [x_K R'_j . x_{K_{l,j}}^{-1}]P = \left[\prod_{i=1, i\neq j}^{n} x_{K_{1,i}} . R'_j\right] P \tag{4}$$

where x_K is Previous Group Key.

3. Also M_1 computes the new key as follows.

$$NLGK = [x_K . R'_j x_{K_{l,j}}^{-1}]P = \left[\prod_{i=1, i\neq j}^{n} x_{K_{l,i}} . R'_j\right] P \tag{5}$$

where x_K is Previous Group Key.

Comparative Analysis of UA-DCGKA

Table 3: Comparative analysis of popular group key agreement protocols

DLP-Protocols		Rounds	Messages	Unicast	Broadcast	Seq exponentions	Seq scalar multiplications
CEGK [18]	Initialize	h	$2m-2$	m	$m-2$	$2h-2$	0
	Join	1	2	1	1	1	0
	Leave	1	1	0	1	$h-1$	0
EGK [14]	Initialize	h	$2m-2$	0	$2m-2$	$2h-2$	0
	Join	1	2	0	2	1	0
	Leave	h	$2(m-1)$	0	$2(m-1)$	$2h$	0
TGDH [10]	Initialize	h	$2m-2$	0	$2m-2$	$2h-2$	0
	Join	2	3	0	3	$3h-3$	0
	Leave	1	1	0	1	$3h-3$	0
STR [19]	Initialize	$m-1$	$2m-2$	0	$2m-2$	$2(m-1)$	0
	Join	2	3	0	3	4	0
	Leave	1	1	0	1	$m-1$	0
GDH.3 [15]	Initialize	$m+1$	$2m-1$	$2m-3$	2	$5m-6$	0
	Join	4	$m+3$	0	$m+3$	$m+3$	0
	Leave	1	1	0	1	m-1	0
ECDLP-based Protocol		Rounds	Messages	Unicast	Broadcast	Seq exponentions	Seq scalar multiplications
GECDH [17]	Initialize	m	m	$m-2$	2	0	$5m-6$
	Join	m	n	0	m	0	$m+3$
	Leave	$m-1$	$m-1$	0	$m-1$	0	$m-1$
TGECDH [17]	Initialize	h	$2m-2$	0	$2m-2$	0	$2h-2$
	Join	2	3	0	3	0	$3h-3$
	Leave	1	1	0	1	0	$3h-3$
DCGKA [our protocol]	Initialize	$m+1$	$2m-1$	$2m-2$	1	0	$2m$
	Join	1	2	1	1	0	6
	Leave	1	1	0	1	0	3

2. $DACGKA^{PP-PKI}$: Paper 2.

- We proposed an ACGKA protocol to generate a group key among the group members. In the process we incorporated authentication in proposed CGKA protocol using **ECDH with PKI** arrangement that binds public keys with respective user identities by means of a Certificate Authority (CA) and proceeds as follows:
- **Defense against man in the middle (MITM) attack:** Diffie Hellman is secured against active attacks by published D-H numbers. Here participants A and B compute their public keys from their own private keys and publish them through PKI . To the extent an intruder can't get in and modify the published public keys, this makes D-H immune to active attacks. It has the additional advantage of eliminating the first two messages of D-H protocol. A and B knowing their own private keys, look for B and A's public keys respectively in PKI and compute their shared key.

$DACGKA^{PP-PKI}$ Continuation

- **Anonymous Authentication using D-H with privacy preserving PKI:** In order to enhance privacy in PKI, recently, Sokjoon Lee et al. proposed a privacy-preserving PKI based on group signature. Using D-H with privacy preserving PKI based on group signature one can provide anonymous authentication so that such *privacy preserving secure attributes such as anonymity, traceability, unlinkability* can be attained.
- **Advantages Using D-H with PKI:** (i) Provides authentication while eliminating MITM attack. (ii) Further it has the additional advantage of reducing $2(m-1)$ messages from the total messages (if PKI is not used) while computing $(m-1)$ ECDH shared keys in first round.

Comparative Analysis of $DACGKA^{PP-PKI}$

DLP-Protocols		*Communication*				*Computation*	*Authentication*	
Protocols		Rounds	Messages	Unicast	Broadcast	Sequential exponentiation	Sequential signatures	Verification
STR [14]	Initialize	$m-1$	$2m-2$	0	$2m-2$	$2(m-1)$	$m-1$	$2(m-1)$
	Join	2	3	0	3	4	2	3
	Leave	1	1	0	1	$m-1$	1	1
GDH.3 [22]	Initialize	$m+1$	$2m-1$	$2m-3$	2	$5m-6$	Not Provided	
	Join	4	$m+3$	0	$m+3$	$m+3$		
	Leave	1	1	0	1	$m-1$		
Dutt's [9]	Initialize	$2m$	$m+1$	0	$m+1$	$3m$	2	$m+1$
	Join	2	$2m-1$	0	$2m-1$	3	2	5
	Leave	2	m	0	m	3	2	$m-1$

ECDLP-Protocols		*Communication*				*Computation*	*Authentication*	
Protocols		Rounds	Messages	Unicast	Broadcast	Sequential Scalar multiplications	Sequential signatures	Verification
GECDH [24]	Initialize	m	m	$m-2$	2	$5m-6$	Not Provided	
	Join	m	m	0	m	$m+3$		
	Leave	$m-1$	$m-1$	0	$m-1$	$m-1$		
TGECDH [24]	Initialize	h	$2m-2$	0	$2m-2$	$2h-2$	Not Provided	
	Join	2	3	0	3	$3h-3$		
	Leave	1	1	0	1	$3h-3$		
DACGKA [Proposed proto-col]	Initialize	2	$m-1$	$m-1$	0	$2m$	*Provided using PKI*	
	Join	1	2	1	1	4		
	Leave	1	$m-2$	$m-2$	0	3		

Conclusion

The proposed protocol being ECDLP based using PKI and contributory in nature, it has all the advantages which are inherent with them such as

- Offering relatively low communication overheads and low computational loads
- Consumption of relatively lower memory storages
- Offering authentication and providing security against active adversary
- Distribution of computational and communicational loads among all the members
- Further, it has the added advantage of dynamically updating the group key without a rerun of the total/entire DACGKA protocol anew, as soon as a member joins or leaves the existing group.

Future Work.

- The proposed protocol provides authentication of the participants using ECDH with Public Key Infrastructure, which may be **difficult in certain environments**. It may be possible to provide authentication using **ECDH integrated signature** scheme for group key agreement, with reduced overall computational and communicational loads.
- Also such security issues as, perfect forward/backward secrecy, replay attack, forgery attack, key compromise impersonation, key control, group forward/backward secrecy, etc. are **yet to be studied** for the proposed protocol.

3. $DACGKA^{Int-Sgn}$: Paper-3

- We propose an ACGKA protocol to generate a GK among the participants of the group. In this technique, a random node acts as the GC that publicly publishes cryptographically strong ECDP (p, a, b, P, n, h) and proceeds as follows:
  Let $M_1, M_2, ..., M_l, ..., M_n$ be the group members and let the GC be M_1.
- **Round.1**: Initially the GC, M_1 forms $(n-1)$ two parties with rest of the nodes M_i, $2 \leq i \leq n$ and performs the authenticated ECDH to generate $(n-1)$ shared keys for $(n-1)$ two parties, as in Table 3:

$DACGKA^{Int-Sgn}$: First Round

Authenticated Two-party key generation

Group Controller-M_1	**Communication**	**Group Member M_i, $2 \le i \le n$**
Long term private: $x_1 \leftarrow [1, n-1]$ $Q_1 = [x_1]P$		Long term private: $x_i \leftarrow [1, n-1]$ $Q_i = [x_i]P$
Short term private: $K_1 \leftarrow [1, n-1]$ $R_1 = [k_1]P$, Signature: $S_1 = k_1^{-1}((Q_1)_x - x_1(R_1)_x)$ Retrieve: $Q_2, R_2, S_2\, Cet(Q_2)$	$Q_1, R_1, S_1, \underline{Cet}(Q_1)$ $Q_2, R_2, S_2, \underline{Cet}(Q_2)$	Short term private: $K_i \leftarrow [1, n-1]$ $R_i = [k_i]P$, Signature: $S_i = k_i^{-1}((Q_i)_x - x_i(R_i)_x)$ Retrieve: $Q_1, R_1, S_1\, Cet(Q_1)$
Verification: $[(R_2)_x]Q_2 + [S_2]R_2 = [(Q_2)_x]P$ Key generation: $K_{1,2} = [k_1]R_2$ $= [k_1]([k_2]P)$ $= [k_1k_2]P$ $= (x_{K_{1,2}}, y_{K_{1,2}}), 2 \le i \le n$		Verification: $[(R_1)_x]Q_1 + [S_1]R_1 = [(Q_1)_x]P$ Key generation: $K_{1,2} = [k_2]R_1$ $= [k_2]([k_1]P)$ $= [k_2k_1]P$ $= (x_{K_{1,2}}, y_{K_{1,2}}), 2 \le i \le n$

Round.2:
Now the GC computes the $(n-1)$ public keys L_i, using authenticated two party shared keys $x_{K_{1,i}}$ generated in Round.1, also choose short term privates, compute short term public and signature as follows and sends it to respective $M_i's$.
Public keys:

$$L_i = [\prod_{j=1, j\neq i}^{n} x_{K_{l,j}}]P, for 2 \leq i \leq n. \tag{6}$$

Short term private:

$$k_l^{'} \leftarrow [1,\ n-1] \tag{7}$$

$$R_l^{'} = [k_l^{'}]P, \tag{8}$$

Signature:

$$S_l^{'} = k_l^{'-1}((Q_l)_x - x_l(R_l^{'})_x) \tag{9}$$

Authenticated messages:

$$M_1 \xrightarrow{L_i, R_l^{'}, S_l^{'}, Cet(L_i)} M_i, for 2 \leq i \leq m. \tag{10}$$

After receiving respective messages, respective members M_i, verifies the authenticated message by checking the equation

Verification:

$$[(R_l^{'})_x]Q_l + [S_l^{'}]R_l^{'} = [(Q_l)_x]P. \tag{11}$$

If the equation is valid, respective M_i, compute the group key K as follows

Authenticated Group Key:

$$K = [x_{K_{l,i}}]L_i = [x_{K_{l,i}}][\prod_{j=1, j\neq i}^{n} x_{K_{l,j}}]P = [\prod_{i=1}^{n} x_{K_{l,i}}]P = (x_K, y_K) \tag{12}$$

The GC consolidates all the shared keys created in round.1 into a GK as follows and becomes a member of the group

$$K = [\prod_{i=1}^{n} x_{K_{l,i}}]P = (x_K, y_K) \tag{13}$$

Hence take x_K as authenticated GK among the group participants.

Comparative analysis of $DACGKA^{Int-Sgn}$

Table: Comparative analysis of popular GKA protocols

ECDLP-Protocols		*Communication*				*Computation*	*Authentication*	
Protocols		*Rounds*	*Messages*	*Uni cast*	*Broad cast*	*Seq Scalarmul tiplication*	*Seq signa tures*	*Verifi cations*
GECDH [(Wang Yong. et al., 2006)]	*Initialize*	*n*	*n*	*n-2*	*2*	*5n-6*	*Not Provided*	
	Join	*n*	*n*	*0*	*n*	*n+3*		
	Leave	*n-1*	*n-1*	*0*	*n-1*	*n-1*		
TGECDH [(Wang Yong. et al.2006;)]	*Initialize*	*h*	*2n-2*	*0*	*2n-2*	*2h-2*	*Not Provided*	
	Join	*2*	*3*	*0*	*3*	*3h-3*		
	Leave	*1*	*1*	*0*	*1*	*3h-3*		
ADCGKA [Our Pro-tocol]	*Initialize*	*2*	*2n-1*	*2n-2*	*1*	*2(n+1)*	*2*	*n*
	Join	*1*	*1*	*1*	*1*	*6*	*1*	*1*
	Leave	*1*	*n-2*	*n-2*	*0*	*3*	*0*	*0*

- In the above, the proposed ECDLP-based DCGKA protocol has been compared with ECDLP-based protocols (GECDH, TGECDH are in fact elliptic curve variants of GDH.3, TGDH respectively) in regard to number of messages, rounds, operations and so on.
- In ECDLP-based protocols additions and scalar multiplications are used instead of multiplications and exponentiations (for DLP-based protocols) respectively. As per less computational cost and reduced storage requirements, ECDLP-based key agreement protocols are adaptable to recourse constrained networks such as MANETS and WSN.
- In view of the comparative analysis in the above Table, the proposed protocol is optimal w.r.t communication and computation costs and also provides similar/the same security level with lesser key sizes. Thus it is relatively the best protocol for secure GKA over recourse constrained networks such as MANETS, and WSN, among DLP and ECDLP based schemes discussed in this paper.

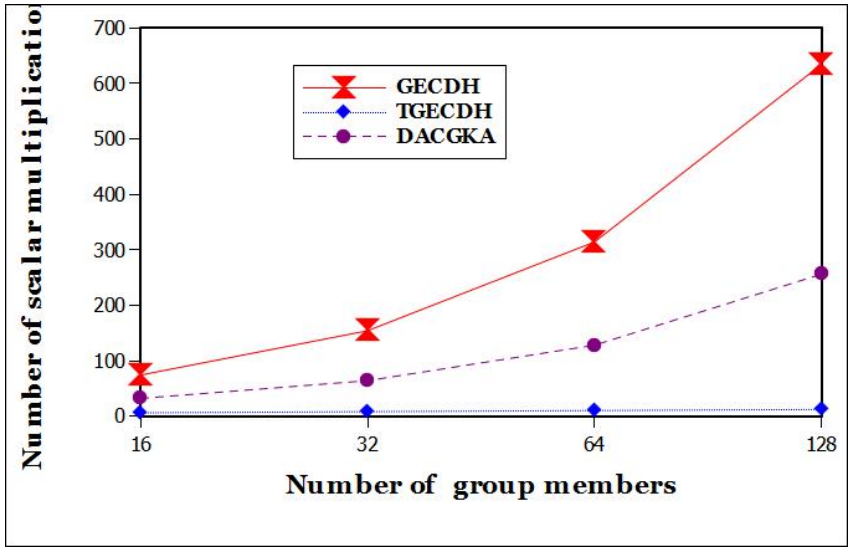
Computational Complexity in terms of individual and total aspects of our ACGKA protocol

Computational Complexity	Sequential Scalar Multiplications in Round-1	Sequential Scalar Multiplications in Round-1	Total
Group Controller (M_1)	$n-1$	$n-1$	$2n-2$
Group Members ($M_i, 2 \leq i \leq n$)	3	1	4
Total	$n+2$	n	$2n+2$

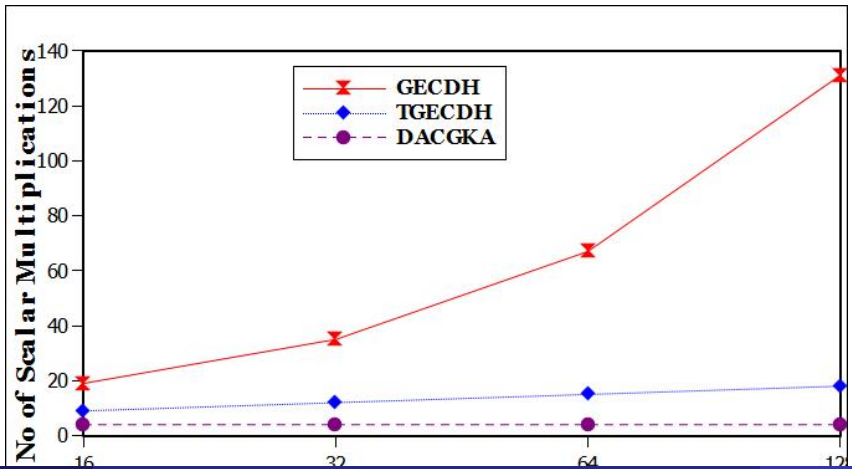
- The proposed protocol being contributory in nature, the computation of the scalar multiplications is equally distributed among all the members except the group controller.
- Although, the group controller has more scalar multiplications to do, it is OK as the proposed protocol being ECDLP-based one, the key sizes are smaller and there are tremendous advances in the computational prowess in recent times.
- In fact, Kazuo S. et al. proposed P-MALU (Parallelized Modular Arithmetic Unit) with which one can achieve over 80K scalar multiplications per second with predictably improved performance.
- For instance, for a large group of size n=40000the propose protocol requires $2n+2$ =80002 sequential scalar multiplication for generating the group key which can be managed within one second time by using P-MALU.

Computational Complexities using graphs

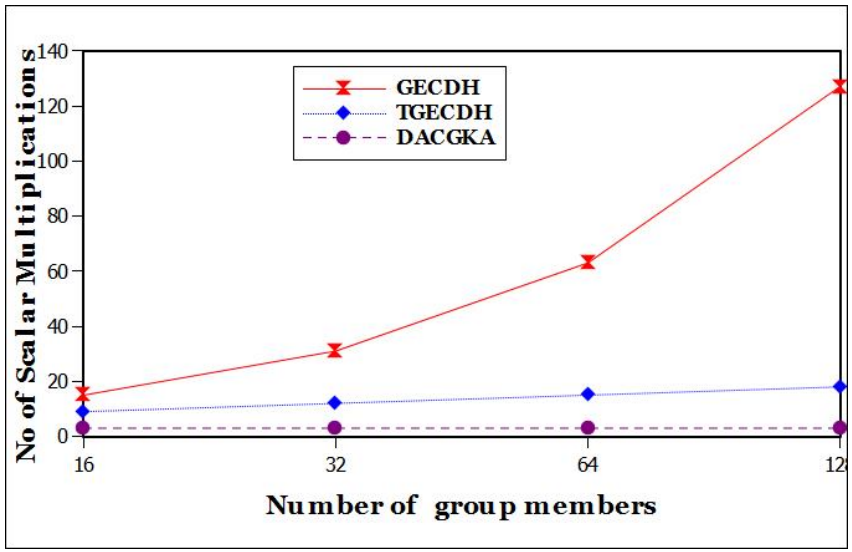
Comparison on scalar multiplications for Initialization of GK:



Comparison on scalar multiplications for Join protocol



Comparison on scalar multiplications for Leave protocol



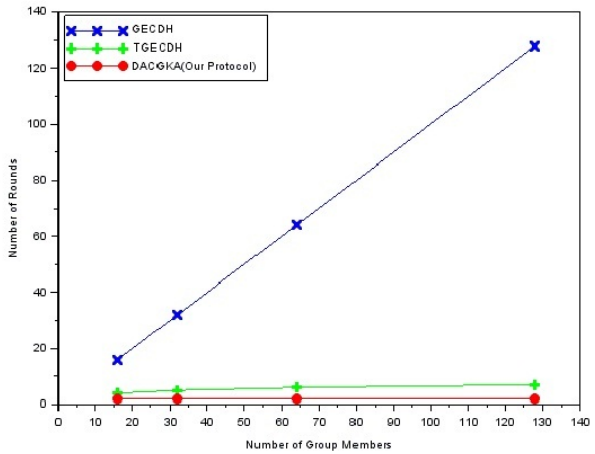
Communication Complexity in terms of individual and total aspects of our ACGKA protocol

Communicational Complexity	*Number of Messages in Round-1*		*Number of Messages in Round-2*		*Total Number of Messages*	
	UniCast	*BroadCast*	*UniCast*	*BroadCast*	*UniCast*	*BroadCast*
Group Controller (M_1)	–	1	$n-1$	–	$n-1$	1
Group Members (M_i, $2 \leq i \leq n$)	1, per each M_i	–	–	–	$n-1$	–
Total	$n-1$	1	$n-1$	–	$2n-2$	1

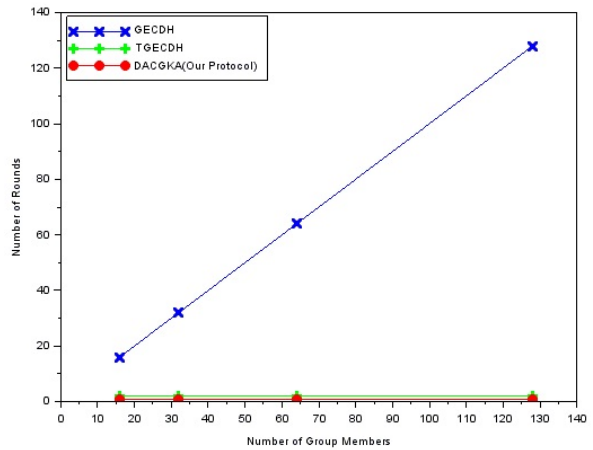
- With unicast-messages, message losses, if any, are **immediately** detectable.
- It is well established that unicast messages are not only less expensive but also less probable to message losses, when compared to broadcast messages.
- In view of above and comparative analysis in the paper making it the best among the dynamic protocols discussed in this paper.

Communication complexities using graphs:

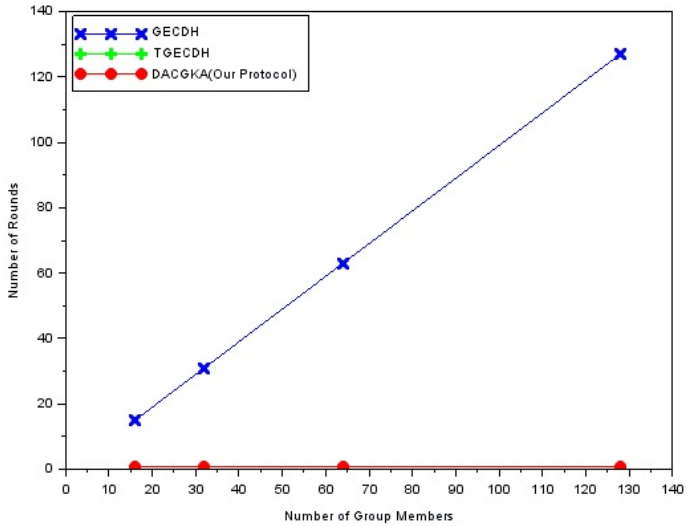
Comparison on number of rounds for Initialization of GK:



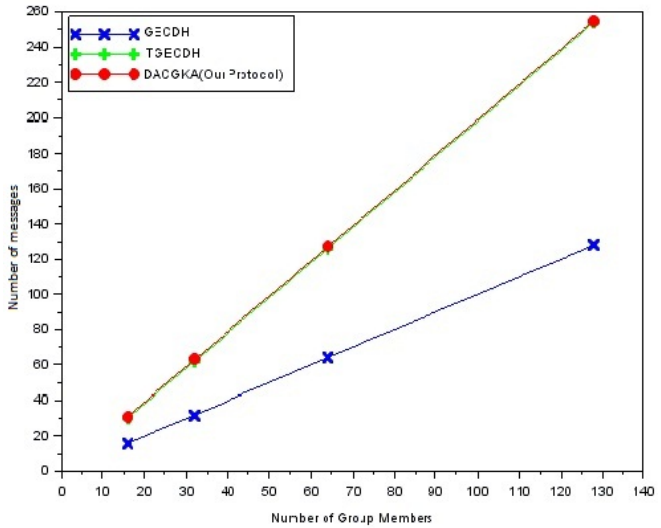
Comparison on number of rounds for Join protocol:



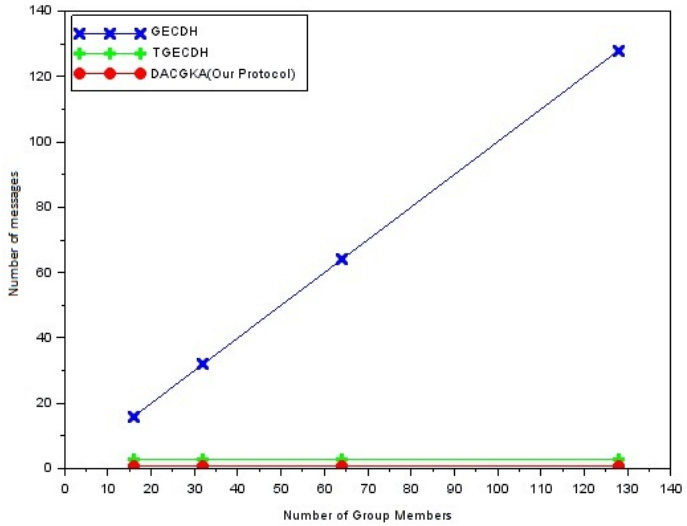
Comparison on number of rounds for Leave protocol:



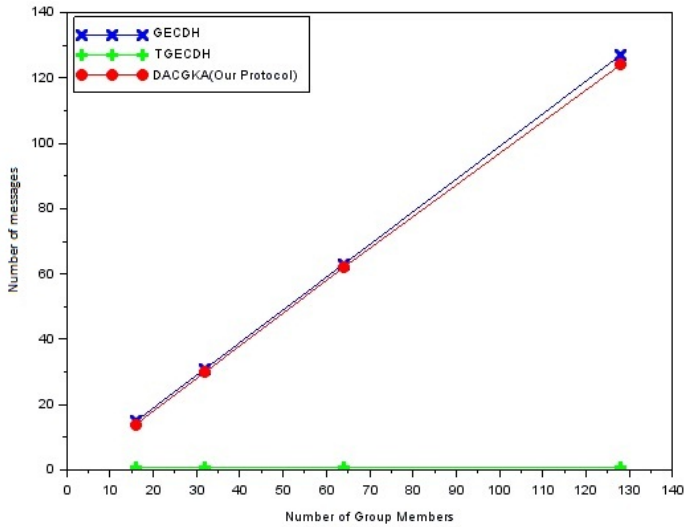
Comparison on number of messages for Initialization of GK:



Comparison on number of messages for Join protocol:

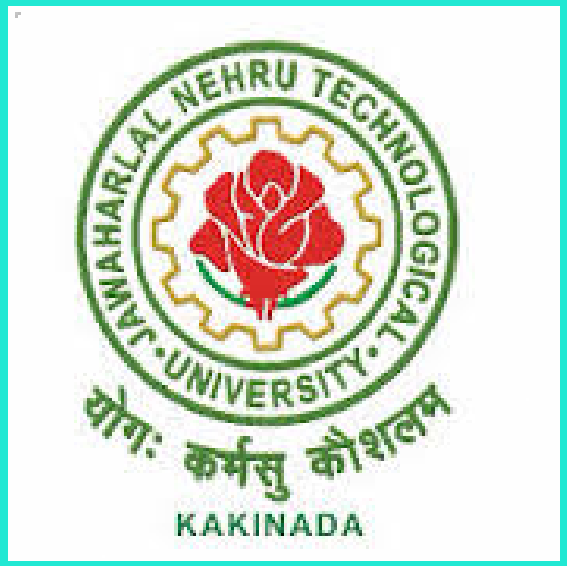


Comparison on number of messages for Leave protocol:



Conclusion:

- In the last paper, we introduced a new DACGKA protocol using ECDH with integrated signature for secure group communication, and analyzed its formal security model and performance.
- Security analysis proves that it is secured against most of the attacks such as key secrecy, replay attack, perfect forward secrecy, key-compromise impersonation, forgery attack, key control, group backward secrecy, and group forward secrecy.
- The theoretical analysis shows that DACGKA is the best protocol in overall performance among the popular DLP and ECDLP based schemes discussed in this paper.
- It offers relatively low communication overheads and lesser computational expense and greater efficiency.
- It requires less memory storage cost than tree based schemes.
- It is suitable for recourse constrained networks such as MANETS and WSN.
- Most importantly its simplicity (i.e) it uses only two rounds which involve simple operations.
- It provides authentication using ECDH with integrated digital signature.
- It does not use any one-way hash function, the security is solely based on hardness of ECDLP



THANK YOU

ANY

QUESTIONS