# A Novel Group Key Management Based on Jacobian Elliptic Chebyshev Rational Map [*]

Qin Ke, Zhou Mingtian, Liu Naiqi, Hao Yujie, Guo Jiandong

School of Computer Science and Engineering,
University of Electronic Science and Technology of China
Chengdu, 610054, P.R.C
yuxuanqk@126.com
{mtzhou, nliu}@uestc.edu.cn

**Abstract.** This paper proposes a novel scheme of group key management based on Jacobian Elliptic Chebyshev Rational Map, named Jacobian Group Key Management(JGKM). The scheme is more efficient than other group key managements since fewer re-keying messages are sent when group membership changes. Besides, it provides both forward and backward secrecy. Therefore, this proposal is helpful to deploy secure multicast over some networks with high latency or limited bandwidth such as wireless network. Furthermore, it fits both small-scale and large-scale groups.

**Keywords:** secure multicast, group key management, Jacobian Elliptic Chebyshev Rational Map.

## 1 Introduction

Encryption is one of the most effective access control mechanisms. All data are encrypted by keys and thus key materials should keep away from attacks. In the context of unicast, a pairwise secure channel is employed to update keys. Contrarily, secure multicast must deal with more complications. Many group key management schemes for secure multicast have been proposed in the past decade [1–5]. These schemes can be classified into three main classes, viz. centralized, decentralized and distributed. Centralized group key management schemes require a single or a small set of entities, named Key Server (KS), to generate or distribute share key to all group members via a secure channel. This kind of schemes can reduce both storage requirement and computational power remarkably. Respectively, decentralized key managements divide the whole group into smaller subgroups. Each subgroup is controlled by a single or several KSs. Distributed schemes allow each member to perform group key generation

[*] Supported by NSF 60473090

and the key generation can either be collaborative or done by a single member. However, no single one can uniquely determine what the key is. Each of the three schemes has its own advantages and disadvantages. No single scheme can fit all applications. For example, Centralized schemes have the risk of single-point-failure while distributed schemes must be weighed against its disadvantages: Many such schemes have a high complexity or a high computational cost. They are best suited for small-scale groups that have nodes with enough computational resources for group Diffie-Hellman key exchange and enough memory to store state information about all of the group's members. There also has to be a trusted mechanism to authenticate the membership join/leave events and the DH public keys.

Most of secure multicast key management schemes focus on decreasing computational and communication overhead [6–8]. In this paper, we introduce a novel approach based on Jacobian Elliptic Chebyshev Rational Map (JECRM) to minimize re-keying messages and computational cost.

JECRM has attracted many researchers from a variety of fields recently, of course cryptographist included[10–13, 23–27]. Some encryption algorithms based on JECRM have been proven not secure[10–13]. In despite of utilizing the same property in this paper, attacks provided in [10] take no effect.

Here is the layout of our paper. Section 2 introduces Jacobian Elliptic Chebyshev Rational Map and its property. Section 3 illustrates Jacobian group key management detailedly. Security and complexity analysis are made in section 4 and 5. Finally, conclusions are drawn in section 6.

## 2 Jacobian Elliptic Chebyshev Rational Map

Jacobian Elliptic Chebyshev Rational Map satisfies chatic properties such as pseudo-random, sensitivity to tiny change of initial conditions, ergodicity, one-way iteration process and etc. All of these properties are interconnected with cryptography closely.

**Definition 1:** Let $v$ be a variable taking value over the interval [-1,1], let $n \geq 2$ be an integer, Jacobian Elliptic Chebyshev Rational Map with modulus $w$ is recursively defined by:

$$R_{n+1}(v,w) = \frac{2vR_n(v,w)}{1 - w^2(1-v^2)(1-R_n^2(v,w))} - R_{n-1}(v,w)$$

where $w \in [0,1]$ and $R_0(v,w) = 1, R_1(v,w) = v$.

JECRM has two important features[24, 25]:

**Theorem 1.** Jacobian Elliptic Chebyshev Rational Map is a One-way function in $v$.

**Theorem 2.** Jacobian Elliptic Chebyshev Rational Map satisfies the semi-group property when $r, s \geq 2$:

$$\begin{aligned} R_{rs}(v, w) &= R_r(R_s(v, w), w) \\ &= R_s(R_r(v, w), w) \end{aligned}$$

There are also other functions that satisfy semi-group property, for example, mod-exp function

$$f_a(f_b(v)) = f_b(f_a(v)) = f_{ab}(v)$$

where $f_a(v) = v^a \mod p$.

The rest of the paper illustrates how to use JECRM to manage group keys in a secure group communication. The reason why we choose JECRM other than mod-exp function is explained in section 4.

## 3 Jacobian Group Key Managemen

The following notations will be used throughout this paper.

| | |
|---|---|
| $v$ | A secret seed selected by KS. None of other members knows $v$ |
| $w$ | A public share selected by KS |
| $u_i$ | The $i^{th}$ user |
| $r_i$ | A large random number selected by KS and delivered to member $u_i$. $r_i$ can be either prime or composite. It makes no difference. Moreover, $r_i$ is known to all members |
| $k_i$ | A pre-placed key encryption key established between $u_i$ and KS |
| $k_{pri}$ | Private key of KS, It is used for signature |
| $k_{pub}$ | Public key of KS. It is used for signature verification |
| $k_{old}$ | The old group key before membership changes |
| $k_{new}$ | The new group key after membership changes |
| $\{M\}_k$ | M is encrypted by $k$ |

Here we make a reasonable assumption: $k_{pub}$ is known to all group members. This can be done using PKI. $k_i$ is a pre-placed key between KS and group members. This can be done using pairwise key $k_{pub}$ and $k_{pri}$.

A group key management must provide re-keying mechanisms when membership changes. JGKM supports the following operations:

-Join: a new member is added to the group.

-Leave: a member is evicted from the group.

-Merge: a subgroup is added to the group.

-Partition: a subgroup is split from the group

-Re-key: the group key must be updated when any above operations occur.

### 3.1 Addition and Merge

**Algorithm 1:**

At the beginning, there is no user in group. The first user is added by following steps.

Step 1. The first user $u_1$ sends joining request to KS.

Step 2. KS responses $u_1$ with $\{R_{r_1}(v, w)\}_{k_1}$, it can only be decrypted by $u_1$.

Step 3. $u_1$ selects $R_{r_1}(v, w)$ as the initial group session key.

Obviously, key distribution starting up needs for only two messages.

With the group growing, more and more users need to be added. JGKM consists of four steps.

**Algorithm 2:**

Step 1. the newcomer $u_{n+1}$ sends joining request to KS.

Step 2. KS responds $u_{n+1}$ with $\{R_{r_{n+1}}(v, w)\}_{k_{n+1}}$. The cipher-text can only be decrypted by $u_{n+1}$.

Step 3. KS multicasts $\{r_1, r_2 \cdots r_{n+1}, w, cmd = addition\}_{k_{pri}}$ to all members. Each member can decrypt $r_i$ $(i = 1, 2 \cdots n + 1)$ and $w$ using $k_{pub}$. $cmd = addition$ indicates that each member should do addition computing after receiving this message. Here we should note that $r_i$ and $w$ are public. Everyone knows $k_{pub}$ can decrypt $r_i$ and $w$. Signature is used here to prove the source of multicast. This can prevent malicious attackers from sending mendacious $r_i, w$ and *cmd*.

Step 4. Newcomer computes the new group key individually according to theorem 1:

$$\begin{aligned} k_{new} &= R_{r_1 \cdots r_n}(R_{r_{n+1}}(v, w), w) \\ &= R_{r_1 \cdots r_{n+1}}(v, w) \end{aligned}$$

Old ones do addition computing, the same new group key is produced by

$$\begin{aligned} k_{new} &= R_{r_{n+1}}(k_{old}, w) \\ &= R_{r_{n+1}}(R_{r_1 \cdots r_n}(v, w), w) \\ &= R_{r_1 \cdots r_{n+1}}(v, w) \end{aligned}$$

Apparently, just three re-keying messages (from step 1 to step 3) are needed in JGKM when member added. Newcomers and old ones can compute new group key without too much interactions. This approach can be extended easily.

If $m$ users, denoted as $u_{n+1}, u_{n+2}, \cdots, u_{n+m}$, will be added:

**Algorithm 3:**

Step 1. $u_{n+1}, u_{n+2}, \cdots u_{n+m}$ send joining requests to KS.

Step 2. KS multicasts $\{R_{r_{n+1}}(v, w)\}_{k_{n+1}}$, $\{R_{r_{n+2}}(v, w)\}_{k_{n+2}}$, $\cdots$, $\{R_{r_{n+m}}(v, w)\}_{k_{n+m}}$. Each member choose the right part to decrypt.

Step 3. KS multicasts $\{r_1, r_2, \cdots, r_{n+m}, w, cmd = addition\}_{k_{pri}}$

Step 4. Newcomer $u_{n+i}$ computes new group key by

$$\begin{aligned} k_{new} &= R_{r_1 \cdots r_{n+i-1} r_{n+i+1} \cdots r_{n+m}}(R_{r_{n+i}}(v, w), w) \\ &= R_{r_1 \cdots r_{n+m}}(v, w) \end{aligned}$$

Old members compute the same group key by

$$\begin{aligned} k_{new} &= R_{r_{n+1} \cdots r_{n+m}}(k_{old}, w) \\ &= R_{r_{n+1} \cdots r_{n+m}}(R_{r_1 \cdots r_n}(v, w), w) \\ &= R_{r_1 \cdots r_{n+m}}(v, w) \end{aligned}$$

Here just $2m + 1$ messages are needed. Now, we can draw a conclusion of JGKM.

**Conclusion 1:** The number of re-keying messages is irrelevant to current group size. It rests with the number of newcomers.

### 3.2 Eviction and Partition

It is easy to cope with member's eviction in JGKM. If member $u_d$ should be evicted, KS multicasts one message:

**Algorithm 4:**

Step 1. KS multicasts $\{u_d, r_d, cmd = eviction\}_{k_{pri}}$ to all members. $cmd = eviction$ indicates that each member should do eviction computing after receiving this message.

Step 2. The rest do eviction computing according below equation:

$$\begin{aligned} k_{new} &= R_{r_1 \cdots r_{i-1} r_{i+1} \cdots r_{d-1} r_{d+1} \cdots r_n}(R_{r_i}(v, w), w) \\ &= R_{r_1 \cdots r_{d-1} r_{d+1} \cdots r_n}(v, w) \end{aligned}$$

The new group key contains no information of $u_d$. Simply, $u_d$ has been removed since he knows nothing about $v$ while $v$ is a secret selected by

KS. Security discussion will be shown in section 4.

It is also easy to extend this approach to fit $m$ users' leaving. Assuming $u_{j_1}, u_{j_2} \cdots u_{j_m}$ leave the group.

**Algorithm 5:**

Step 1. KS multicasts a combined message $\{(u_{j_1}, r_{j_1}), (u_{j_2}, r_{j_2}), \cdots, (u_{j_m}, r_{j_m}), cmd = eviction\}_{k_{pri}}$ to entire group.

Step 2. The rest do eviction computing and new group key is given by:

$$k_{new} = R_{r_1 \cdots r_{i-1} r_{i+1} \cdots r_n}(R_{r_i}(v, w), w), (i \neq j_1, j_2 \cdots j_m)$$

Similarly, the new group key contains no information of $u_{j_1}, u_{j_2}, \cdots, u_{j_m}$ Now, we have another conclusion:

**Conclusion 2:** Only one re-keying message is sent when someone evicted.

## 4 Security Analysis

The security of group key management protocols can be measured by:

-Backward secrecy: new members should not be able to read past traffic.

-Forward secrecy: Former members should not be able to read present and future traffic.

-Collusion attack: Evicted members must not be able to work together and share their individual piece of information to regain access to the group key.

From section 3, we know that JGKM's security bases on the secrecy of parameter $v$. JGKM provides both forward and backward secrecy grounding on the fact that $R_n(v, w)$ is one-way function in $v$ and sensitive to initial condition $v$.

Considering an adversary $u_d$ a group insider, he knows $r_i$ $(i = 1, 2...n)$ and $R_d(v, w)$. He does not know $v$ and $R_{r_i}(v, w), r_i \neq d$. The new group key has changed into $k_{new}$ after its leaving.

**Discussion 1:** Section 3.2 indicates that

$$\begin{aligned} k_{new} &= R_{r_1 \cdots r_{i-1} r_{i+1} \cdots r_{d-1} r_{d+1} \cdots r_n}(R_{r_i}(v, w), w) \\ &= R_{r_1 \cdots r_{d-1} r_{d+1} \cdots r_n}(v, w) \end{aligned}$$

This equation does not contain any information of $u_d$ at all. If $u_d$ want to regain the new group key, he must know $r_i, v, w$. But $v$ is a secrecy selected by KS. $u_d$ can't resolve $v$ from $R_d(v, w)$ since $R_d(v, w)$ is a one-way function in $v$.

**Discussion 2:** At the same time, section 3.2 implies

$$k_{old} = R_{r_d}(k_{new}, w)$$

Obviously, $k_{old}$ is also a one-way function in $k_{new}$, resolving $k_{new}$ from above equation is a hard problem as well. Not any efficient method or quantitative measurement [21, 22] have been found to finish this attack.

Similarly, If $u_d$ is a newcomer, it is also impossible to recover $k_{old}$ by

$$k_{new} = R_{r_d}(k_{old}, w)$$

However, we must point out that $r_i$ should be restricted. Otherwise an adversary may recover group key without effort.

**Restriction 1:** $r_i \neq 1$

If $r_i = 1$, member $u_i$ will receive $R_1(v, w)$ according to algorithm 1 and algorithm 2. On the other hand, according to definition 1, $R_1(v, w) = v$. This indicates member $u_i$ receives the secret parameter $v$. That is forbidden.

**Restriction 2:** $r_i$ must be larger enough, e.g. $r_i \geq 2^{32}$

According to definition 1, if $r_i$ is very small, e.g. $r_i = 2$ *or* 3, it is possible to resolve $v$ from $R_{r_i}(v, w)$. In order to enhance the security of JGKM, we choose random number $r_i \geq 2^{32}$.

**Restriction 3:**

$$r_i \neq \prod_{\substack{j \in [1,n] \\ j \neq i}}^{l} r_j, \quad (l = 1, 2 \cdots n - 1)$$

If $r_n = r_1 r_2 \cdots r_{n-1}$, according to algorithm 2, $u_n$ will receive KS's response $R_{r_n}$

$$R_{r_n}(v, w) = R_{r_1 \ldots r_{n-1}}(v, w)$$

which is the new group key after $u_n$'s leaving.

Furthermore, restriction 3 also eliminates collusion attack.

Now, we will explain why JECRM is chosen other than mod-exp function. Although mod-exp function does satisfy semi-group property, it is a one-way function in $n$ rather than $v$. That is , it is easy to resolve $v$ from mod-exp function. Therefore, $v$ can't be chosen as common secrecy. If mod-exp function is chosen, it is obvious that an attacker can recover $k_{old}$ after his addition or regain $k_{new}$ after his eviction.

## 5 Complexity Analysis

JGKM involves multiple floating-point operations. We employ unique decomposition theorem to improve JGKM's efficiency. According to unique decomposition theorem, an integer $r$ can be uniquely decomposed to $r = p_1^{l1} p_2^{l2} \cdots p_n^{ln}$ where $p_i$ are primes. If $r$ is not decomposed, we must do $r$ iterations in order to computer $R_r(v, w)$. On the other hand, according to following equation, at most $p_1 l_1 + p_2 l_2 + \cdots p_n l_n$ floating-point operations are needed.

$$R_r(v,w) = R_{p_1^{l1} p_2^{l2} \cdots p_n^{ln}}(v,w) = \underbrace{R_{p_1} \cdots}_{l_1}(\underbrace{R_{p_2} \cdots}_{l_2}(\underbrace{R_{p_n} \cdots R_{p_n}}_{\cdots \quad l_n}(v,w),w),w)$$

For example, we choose $r = 7^5 17^3 31^4 67^{10} \approx 2^{128}$ (The key length of AES is 128 bits), thus computation of $R_r(v, w)$ need only $7 \times 5 + 17 \times 3 + 31 \times 4 + 67 \times 10 = 880$ other than $2^{128}$ floating-point operations. At present, Chinese Godson-2 can perform 2 billion single-precision floating-point and 1 billion double-precision floating-point operations per second. That is, Godson-2 perform $R_r(v, w)$ within $0.13ms$. It should be much faster in mainframe computers.

## 6 Summary

In this paper, we proposed a group key management based on Jacobian Elliptic Chebyshev Rational Map. On one hand, it has the structure of centralized schemes, on the other hand, it has the virtue of distributed schemes. Briefly, it has following features:

-A Key Server is involved in performing initialization.

-Each group member contributes its share $r_i$ to group key.

-Each member receives a secrecy from KS.

-Each member make use of others' share and his own secrecy to compute new group key without interaction.

-As the group grows, old members' secrecy remain unchanged. New members' secrecy are sent by pre-placed secret channel.

-As the group shrinks, departing members' secrecy are removed from the new key.

-JGKM does not need any auxiliary keys except for pre-placed keys.

-JGKM is more efficient than other schemes[5–7, 14, 15]. The number of re-keying messages is irrelevant to current group size. Only $2m + 1$ re-keying messages are sent when group grows and only 1 re-keying messages are sent when group shrinks. This makes it differ from other schemes. Most of group key managements involve $2m + \log(n)$ re-keying messages where $n$ is current group size.

## Acknowledgment

We'd like to thank MSEC and RMT working group of IETF for their helpful discussions.

## References

1. A. Ballardie. Scalable multicast key distribution. RFC1949, 1996
2. T. Hardjono, Verisign. The Multicast Group Security Architecture. RFC3740. 2004
3. D.Wallner, E.Harder, R.Agee. Key Management for Multicast: Issues and Architectures RFC2627. June 1999
4. Fan Du, Lionel M. Ni, Abdol-Hossein Esfahanian . Toward Solving Multicast Key Management Problem . Computer Communications and Networks, 1999. P232-P236
5. Sandro Rafaeli, David Hutchison. A Survey of Key Management for Secure Group Communication.ACM Computing Surveys, Vol. 35, No. 3. September 2003. P309-P329
6. Claudiu Duma, Nahid Shahmehri, Patrick Lambrix . A Hybrid Key Tree scheme for Multicast to Balance Security and Efficiency Requirements. Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. P378-P383
7. Josep Pegueroles, Francisco Rico-Novella. Reducing Latency in Multicast Group Re-keying using Eulers Theorem. IEEE Communication letters, Vol.6, NO.4, April 2003. P128-133
8. Ran.Canneti. Efficient Communication-Storage Tradeoffs For multicast Encryption. Advances in Cryptology-EUROCRYPT 1999, P456-477
9. Mingyan Li, R. Poovendran, C. Berenstein. Design of Secure Multicast Key Management Schemes With Communication Budget Constraint. IEEE COMMUNICATIONS LETTERS, VOL. 6, NO. 3, MARCH 2002,
10. Ljupco Kocarev, Zarko Tasev. Public-Key Encryption Based On Chebyshev Mnnaps. Circuits and Systems, ISCAS'03, Vol 3, P28-31. 2003
11. Zhou Hong, Yu Jun, Ling Xie-Ting. Design of chaotic feed forward stream cipher. ACTA ELECTRONICA SINICA. Vol.26 No.1 Jan 1998, P122-P126
12. Gonzalo Alvarez. Security problems with a chaos-based deniable authentication scheme. arXiv:nlin. CD/0412023 v1 9 Dec 2004
13. Pina Bergamo, Paolo D'Arco, Alfredo De Santis and Ljupco Kocarev. Security of Public Key Cryptosystems based on Chebyshev Polynomials. arXiv:cs.CR/0411030 v1 10 Nov 2004
14. Mnnichael Steiner, Gene Tsudik, Michael Waidner. Diffie-Hellman Key Distribution Extended To Group Communication. Proceedings of the 3rd ACM conference on Computer and communications security, New Delhi, India. 1996. P31-37
15. Yair Amir, Yongdate Kim, Cristina Nita-Rotaru, Gene Tsudik. On The Performance of Group Key Agreement Protocols. ACM Transactions on Information and System Security, Vol. 7, No. 3, August 2004, P 457-P488.

16. William Aiello, Steven M.Bellovin, Mattblaze. Just Fast Keying: Key Agreement in a Hostile Internet. ACM Transactions on Information and System Security, Vol.7, No.2, May 2004, P242-P273
17. Tao Yang. A survey of chaotic secure communication systems. International Journal of Computational Cognition. Volume 2, Number 2, June 2004 P81-P130,
18. David A.McGrew, Alan T.Sherman. Key Establishment in Large Dynamic Groups Using One-Way Function Trees. IEEE Computer Society Vol. 29, No.5, MAY 2003 P444-P458
19. Wei-Chi Ku, Shuai-Min Chen. An Improved Key Management Scheme for Large Dynamic Groups Using One-Way Function Trees. Proceedings of the 2003 International Conference on Parallel Processing Workshops, 2003. P245-251
20. Y. Kim. A. Perrig and G. Tsudik. Simple and Fault-Tolerant Key Agreement for Dynamic Collaborative Groups. In Proceedings of the 7th ACM Conference on Computer and Communications Security (ACM CCS 2000) Nov. 2000, P235-244.
21. M. Abramowitz and I. A. Stegun. Handbook of Mathematical Functions. Dover Publications, 1970
22. E.L.Wachspress. Evaluating Elliptic Functions and Their Inverses. Computers and Mathematics with Applications. NO.39. 2000. P230-236
23. B.C. Carlson. Jacobian elliptic functions as inverses of an integral. Journal of Computational and Applied Mathematics 174(2005) P355-P359.
24. Aya Kato and Tohru Kohda. Solvable 2-dimensional Rational Chaotic Map Defined by Jacobian Elliptic Functions. Circuits and Systems, 2005. P1477-P1480
25. T. Kohda, H. Fujisaki. Jacobian elliptic Chebyshev rational maps. Physica D 148 (2001) P242-P254
26. Marco Gotz, Kristina Kelber, and Wolfgang Schwarz. Discrete-Time Chaotic Encryption SystemsPart I: Statistical Design Approach. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMSI: FUNDAMENTAL THEORY AND APPLICATIONS, VOL. 44, NO. 10, OCTOBER 1997. P963-P970
27. Junghyun Nam, Jinwoo Lee, Seungjoo Kim, Dongho Won. DDH-based group key agreement in a mobile environment. The Journal of Systems and Software, 2005 P73-P83