A SYMMETRIC GROUP-BASED PUBLIC-KEY CRYPTOSYSTEM WITH SECRET PARTITION-DEPENDENT DECRYPTION

KAVEH DASTOURI

Abstract. We present a purely theoretical public-key cryptosystem based on the symmetric group S_n and a one-way function derived from conjugacy class sizes. The secret key is a carefully chosen partition $\lambda \vdash n$, and the public key is $f(\lambda) = |C_\lambda| \cdot m_1(\lambda)$. Decryption inherently requires knowledge of λ to compute $\phi(f(\lambda))$ or equivalently to factor $f(\lambda)$. The system combines combinatorial inversion hardness and integer factorization difficulty, ensuring that only someone who knows λ can decrypt. Historical context, worked examples, and theoretical security analysis are included.

1. Introduction

Public-key cryptography was initiated by Diffie and Hellman [3] and realized in practice via the RSA scheme [9], which relies on the difficulty of factoring large integers. Beyond number-theoretic approaches, algebraic and group-theoretic cryptography has been studied, including braid groups [8, 2], non-commutative schemes [1, 4], and symmetric group-based constructions [5, 6].

Symmetric groups possess rich combinatorial structure, including partitions and conjugacy classes. Previous works [6] showed that functions based on conjugacy class sizes can be one-way. In this paper, we propose a cryptosystem in which decryption is only possible with knowledge of the secret partition λ, providing a theoretical foundation where combinatorial and number-theoretic hardness are intertwined.

2. Preliminaries

The symmetric group S_n, the set of all permutations of $\{1, \dots, n\}$, is fundamental in mathematics, linking algebra, combinatorics, and computer science. Let S_n denote the symmetric group on n letters. A partition $\lambda = (\lambda_1, \dots, \lambda_\ell) \vdash n$ defines the cycle type of a permutation. The conjugacy class corresponding to λ is

$$C_\lambda = \{\sigma \in \mathrm{S}_n : \text{cycle type}(\sigma) = \lambda\},$$

with size

$$|C_\lambda| = \frac{n!}{\prod_i (\lambda_i^{m_i} m_i!)},$$

where m_i is the multiplicity of the part i.

Define the one-way function:

$$f(\lambda) = |C_\lambda| \cdot m_1(\lambda),$$

where $m_1(\lambda)$ is the number of 1-cycles. Computing $f(\lambda)$ is polynomial-time, but recovering λ from $f(\lambda)$ is combinatorially hard. Actually, $f(\lambda)$ is the permutation character (See [7, Chapter 13])

3. Partition Selection for Hard-to-Factor $f(\lambda)$

To make $f(\lambda)$ hard to factor, we choose partitions with:

(1) Exactly one 1-cycle ($m_1(\lambda) = 1$), preserving large primes in $n!$.
(2) Remaining parts as distinct composition numbers ($\lambda_2, \ldots, \lambda_\ell > 1$), preventing repeated primes in the denominator.
(3) Optional: parts as products of small primes to keep the denominator manageable.

Proposition 3.1. *Partitions chosen this way ensure $f(\lambda)$ contains large prime factors, making factorization computationally infeasible for large n.*

Sketch. Since $n!$ contains all primes up to n and the denominator only cancels small primes from repeated parts, large primes survive in $f(\lambda)$. Factorization without knowing λ is equivalent to factoring a large integer with unknown prime composition, which is hard. □

4. Worked Example

Let $n = 20$ and choose

$$\lambda = (1, 3, 4, 5, 7), \quad \sum_i \lambda_i = 20.$$

Then

$$|C_\lambda| = \frac{20!}{1! \cdot 3 \cdot 4 \cdot 5 \cdot 7 \cdot (1!^5)} = \frac{20!}{420}, \quad f(\lambda) = |C_\lambda| \cdot 1 = \frac{20!}{420}.$$

Large primes 19, 17, 13, 11 survive in $f(\lambda)$, demonstrating why factorization is hard without λ.

5. Cryptosystem Design

5.1. Key Generation.

(1) Choose a large n and construct a partition $\lambda \vdash n$ as above.
(2) Compute $f(\lambda)$.
(3) Factor $f(\lambda)$ using knowledge of λ to compute $\phi(f(\lambda)) = \prod_i p_i^{e_i - 1}(p_i - 1)$.
(4) Choose encryption exponent e coprime to $\phi(f(\lambda))$ and compute $d = e^{-1} \mod \phi(f(\lambda))$.
(5) **Public key**: $(f(\lambda), e)$, **Secret key**: λ (or equivalently d and factorization of $f(\lambda)$).

5.2. Encryption.

$$c = m^e \mod f(\lambda), \quad m \in [1, f(\lambda) - 1].$$

5.3. Decryption (requires λ).

(1) Using λ, factor $f(\lambda)$ and compute $\phi(f(\lambda))$.
(2) Compute $d = e^{-1} \mod \phi(f(\lambda))$ if not precomputed.
(3) Recover $m = c^d \mod f(\lambda)$.

Remark 5.1. *Without knowledge of λ, one cannot factor $f(\lambda)$ or compute $\phi(f(\lambda))$. Therefore, decryption inherently requires the secret partition, linking combinatorial and number-theoretic hardness.*

6. Security Considerations

- One-way function: $f(\lambda)$ is easy to compute but hard to invert (recover λ).
- Hard-to-factor: "one 1 + distinct compositions" preserves large primes.
- Decryption requires secret: Knowledge of λ is necessary to factor $f(\lambda)$ and compute $\phi(f(\lambda))$.

7. Extensions

- Use multiple partitions $\lambda_1, \ldots, \lambda_k$ to increase hardness.
- Embed messages using symmetric group characters or group algebra methods.
- Extend to other non-abelian groups with complex conjugacy structures.

8. Conclusion

We presented a symmetric group-based public-key cryptosystem where decryption explicitly requires knowledge of the secret partition λ. The system combines combinatorial hardness of inverting $f(\lambda)$ with the number-theoretic hardness of factoring, yielding a purely theoretical cryptosystem with clear dependence on the secret.

References

[1] M. Anshel, M. Anshel, D. Goldfeld, *An algebraic method for public-key cryptography*, Math. Res. Lett., 6 (1999), no. 3–4, 287–291. DOI: 10.4310/MRL.1999.v6.n3.a3
[2] J. H. Cheon, B. Jun, *A polynomial time algorithm for the braid Diffie-Hellman conjugacy problem*, CRYPTO 2003, LNCS, vol. 2729, Springer, 2003, pp. 212–225. DOI: 10.1007/978-3-540-45146-$4_1$3
[3] W. Diffie, M. E. Hellman, *New directions in cryptography*, IEEE Trans. Inform. Theory, 22 (1976), no. 6, 644–654. DOI: 10.1109/TIT.1976.1055638
[4] A. Myasnikov, V. Shpilrain, A. Ushakov, *Non-commutative Cryptography and Complexity of Group-theoretic Problems*, Math. Surveys Monogr., vol. 177, Amer. Math. Soc., 2011.
[5] A. G. Myasnikov, V. Shpilrain, A. Ushakov, *Group-based Cryptography*, Adv. Courses Math. CRM Barcelona, Birkhäuser, 2008. DOI: 10.1007/978-3-7643-8827-0
[6] A. Isacsson, *A hashing algorithm based on a one-way function in the symmetric group*, Master's thesis, KTH Royal Institute of Technology, 2022. http://www.diva-portal.org/smash/get/diva2:1662325/FULLTEXT01.pdf
[7] G. James, M. Liebeck, *Representations and Characters of Groups*, 2nd ed., Cambridge University Press, 2001.
[8] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. Kang, C. Park, *New public-key cryptosystem using braid groups*, CRYPTO 2000, LNCS, vol. 1880, Springer, 2000, pp. 166–183. DOI: 10.1007/3-540-44598-$6_1$0
[9] R. L. Rivest, A. Shamir, L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM, 21 (1978), no. 2, 120–126. DOI: 10.1145/357980.358017