Multi graph-based cryptographic algorithm

Improved Perfect Code Cryptosystem

류지은	국민대
염용진	국민대
윤승태	CSHL
강주성	국민대
김용빈	국민대

Brief History of Developing IPCC

Combinatorial cryptogr

PMDF(Perfect Minus D

Revisiting Koblitz's wor

IPCC(Improved Perfect

윤승태
CSHL(Cold Spring Harbor Laboratory)

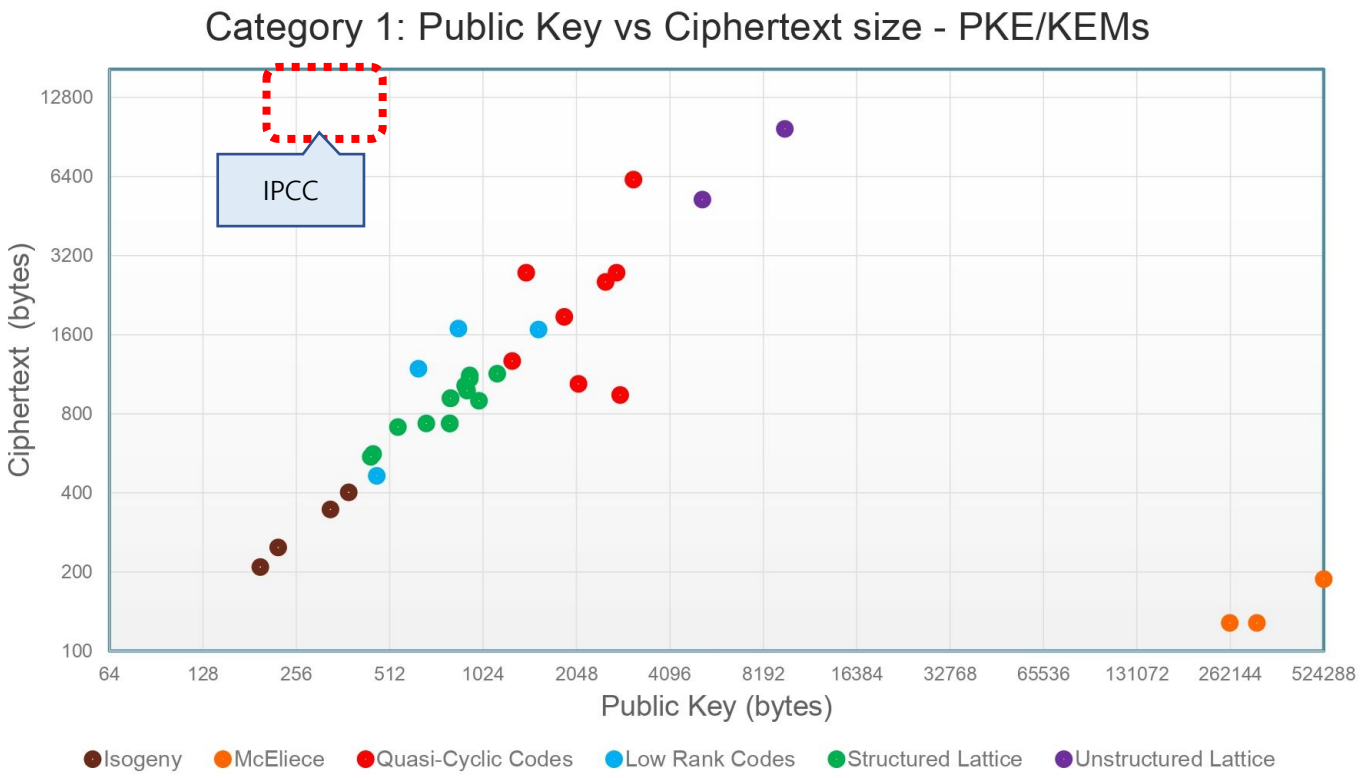
Attack on IPCC by T. Lange (2022)

Countermeasure for Lange's attack and updating IPCC

Comparison of IPCC and NIST PQC candidates

Features of IPCC

- Small public key but huge ciphertext
- Fast decryption (parallelizable)
- Suitable for whitebox cryptography (one-wayness) with large memory

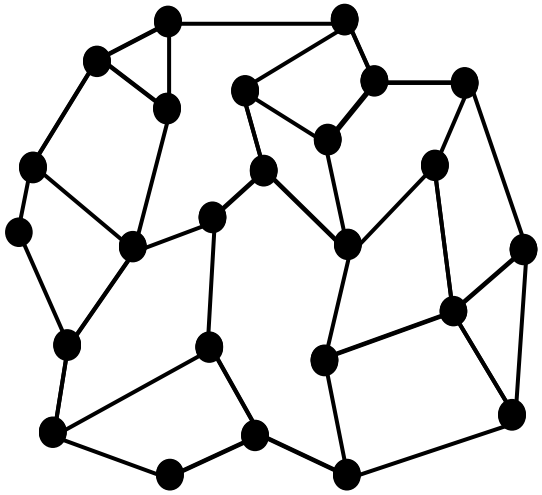


Hard problem in Graph-based Public Key Cryptosytem

Hard problem in RSA (Integer factoring)

Given N → difficult → Find p and q s.t. $N = pq$

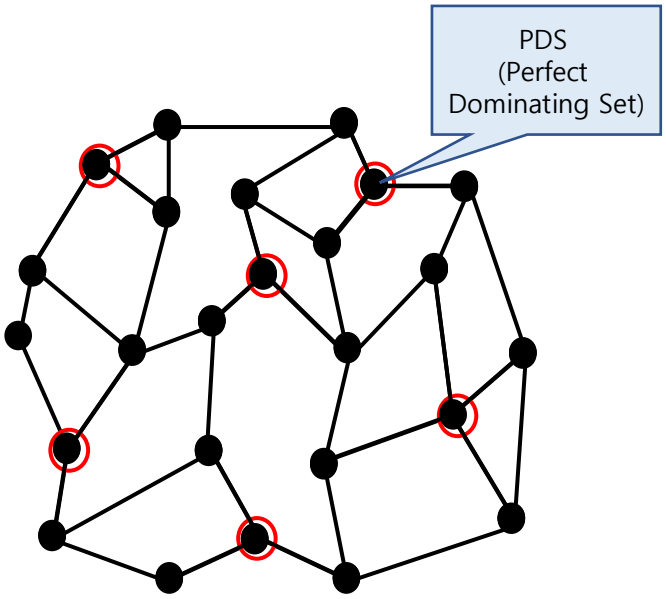
A Hard problem in Graph theory



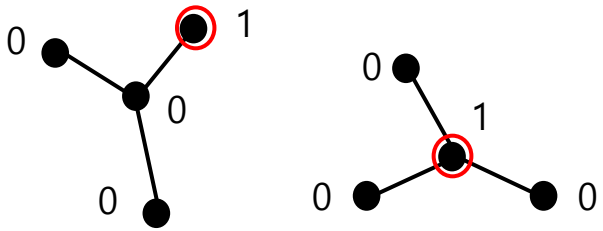
graph $G = (V, E)$

difficult

(NP-hard)



Find a PDS of G.

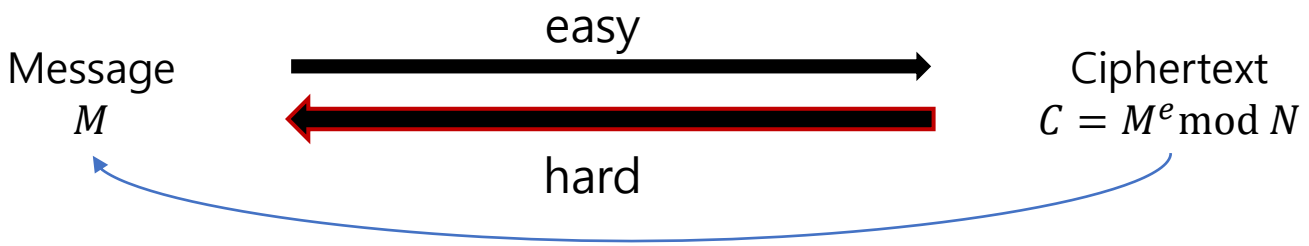


Invariant polynomial (of degree one)

$$p(\cdot) = x_{v_1} + x_{v_2} + x_{v_4} + x_{v_6} = 1$$

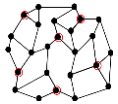
Trapdoor one-way function

RSA



IPCC

public key: $G = (V, E)$ 

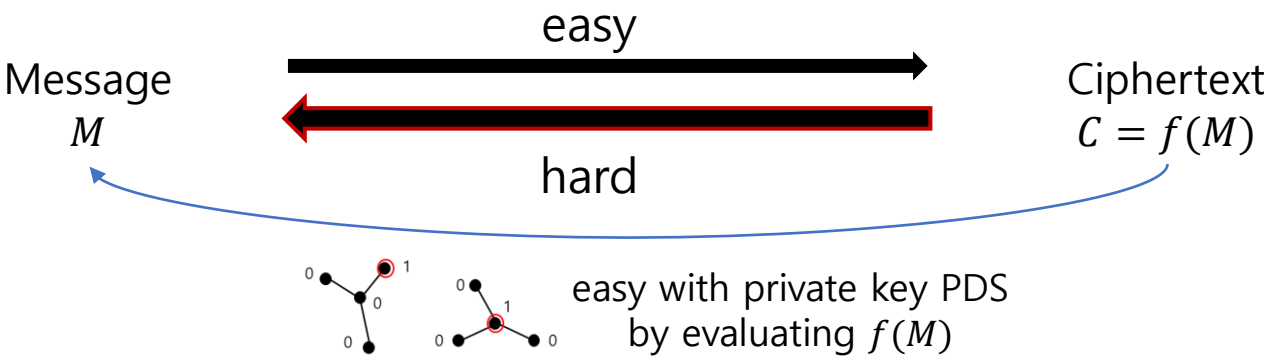
private key: PDS of G 

$$M = m_1 + m_2 + \cdots + m_n$$

$$m_1 p_{11}(\cdot) p_{12}(\cdot) + m_2 p_{21}(\cdot) p_{22}(\cdot) + \cdots + m_n p_{n1}(\cdot) p_{n2}(\cdot)$$

Reduction

$$C = f(M)$$



Cryptanalysis of IPCC (Lange)

- IPCC exposes sub-messages m_i
  caused by insufficient mixing (sparse polynomial of low degree).

$$m_1 p_{11}(\cdot)p_{12}(\cdot) + m_2 p_{21}(\cdot)p_{22}(\cdot) + \cdots + m_n p_{n1}(\cdot)p_{n2}(\cdot)$$

- Lange's attack is more efficient than plaintext recovery attack (dedicated attack).

Analysis and Our Countermeasures

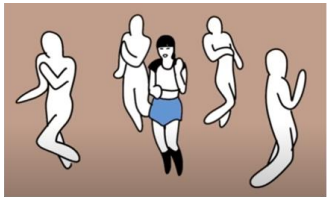
- We need new strategy to hide sub-messages.
- **Invariant polynomials of higher degree** solve the problem.

  (We can construct an algorithm for selecting polynomials on the fly)

- And more ideas we can still try …

It may take longer than expected but IPCC will be antifragile.





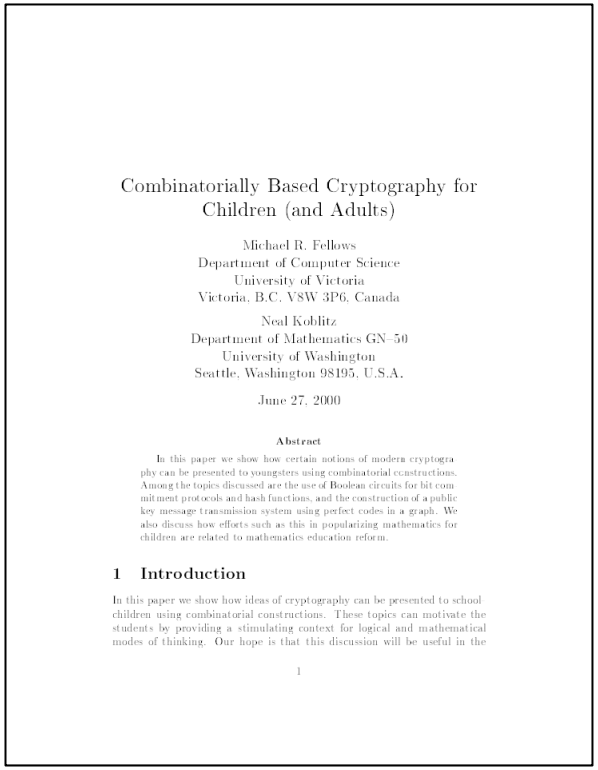
INDEX

Purpose of research about Perfect Code Cryptosystems

In 1992, Koblitz and Fellows proposed
a graph-based public key cryptosystem

This cryptosystem achieves one-wayness by leveraging
the complexity of finding a certain subgraphs in a graph,
and it is predicted to be an NP-hard problem
that has potential applications in PQC

Despite its potential as a post-quantum cryptosystem,
the cryptosystem has not received much attention
due to its slow encryption speed and high memory usage,
which increase as the security level is raised



Combinatorially Based Cryptography for Children (and Adults)

Michael R. Fellows
Department of Computer Science
University of Victoria
Victoria, B.C. V8W 3P6, Canada

Neal Koblitz
Department of Mathematics GN–50
University of Washington
Seattle, Washington 98195, U.S.A.

June 27, 2000

Abstract

In this paper we show how certain notions of modern cryptography can be presented to youngsters using combinatorial constructions. Among the topics discussed are the use of Boolean circuits for bit commitment protocols and hash functions, and the construction of a public key message transmission system using perfect codes in a graph. We also discuss how efforts such as this in popularizing mathematics for children are related to mathematics education reform.

1 Introduction

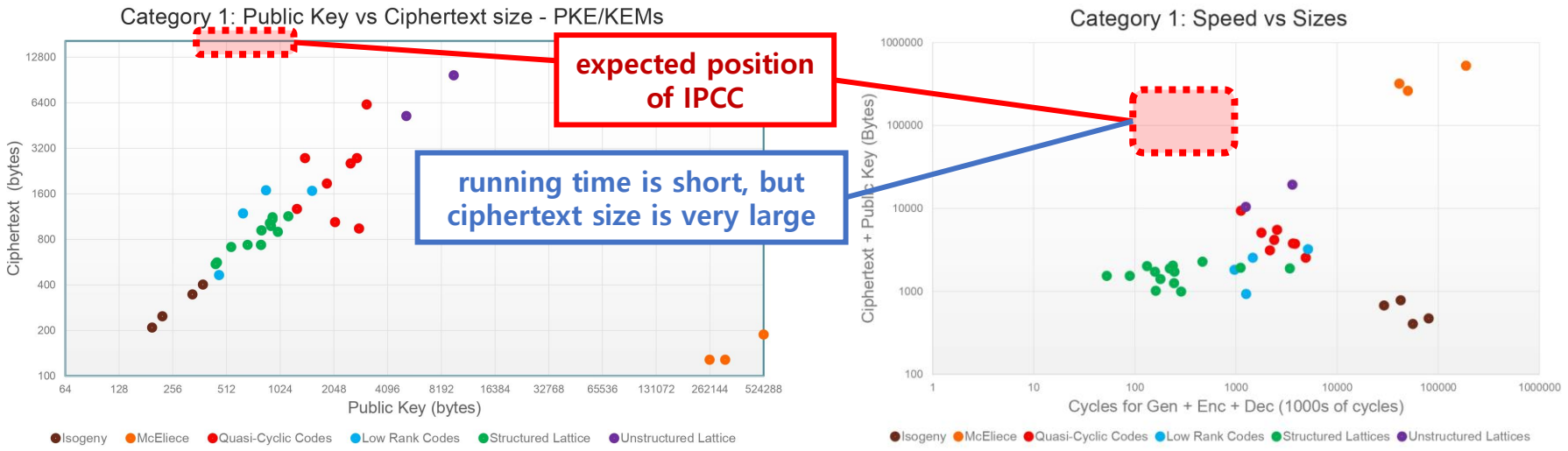
In this paper we show how ideas of cryptography can be presented to school children using combinatorial constructions. These topics can motivate the students by providing a stimulating context for logical and mathematical modes of thinking. Our hope is that this discussion will be useful in the

1

M. Fellows, and N. Koblitz, "Combinatorially based cryptography for children (and adults)," Congressus Numerantium, 9-9, 1994.

Purpose of research about Perfect Code Cryptosystems

The improved cryptosystem is expected to exhibit different characteristics than the algorithms proposed as NIST PQC standardization candidates, and it may be more effective at providing one-wayness or whitebox cryptographic encoding in a memory-rich environment



Characteristics of NIST PQC standardization candidate algorithms

D. Moody, "The 2nd round of the NIST PQC standardization process", In the second PQC standardization Conference, 2019.

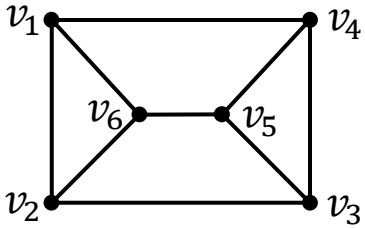
Notation

V	: set of vertices
E	: set of edges, $E \subseteq V \times V$
$G=(V,E)$	: graph generated by V and E
$N[v]$	: set of neighboring vertices of v, including itself
$P^*(V)$	: power set of a set V excluding the empty, $P(V)\backslash\{\emptyset\}$

3-regular graph

If a graph $G = (V, E)$ satisfies the following,
it is called a 3-regular graph

$$\forall v \in V,\ |N[v]| - 1 = 3$$

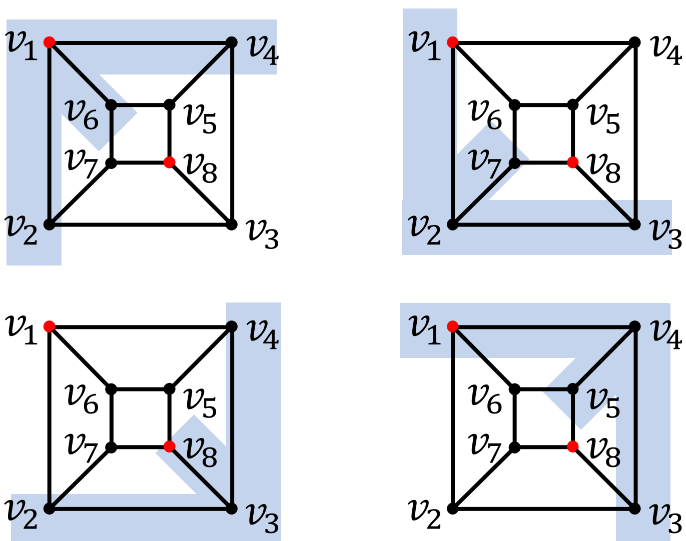


example of 3-regular graph

PDS (Perfect Dominating Set)

For every vertex v in the graph $G=(V,E)$,
if $N[v]$ contains exactly one of the elements of vertex set $A \subseteq V$,
then A is called the PDS of G.

$$\exists A \subset V \ \text{s.t.}\ \forall v \in V, \qquad |N[v] \cap A| = 1$$



Every elements v of A belong to the set $N[v]$ only once

PDS

$A = \{v_1, v_8\}$

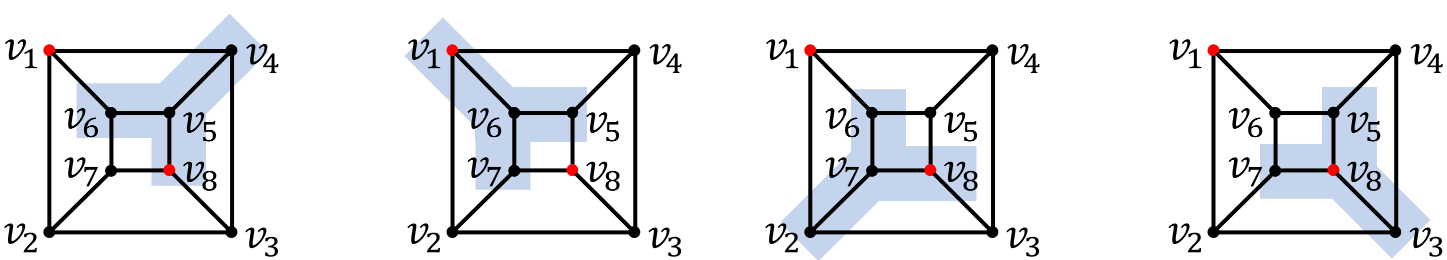
$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$

$N[v_1] = \{v_1, v_2, v_4, v_6\}$ $\qquad N[v_2] = \{v_1, v_2, v_3, v_7\}$

$N[v_3] = \{v_2, v_3, v_4, v_8\}$ $\qquad N[v_4] = \{v_1, v_3, v_4, v_5\}$

$N[v_5] = \{v_4, v_5, v_6, v_8\}$ $\qquad N[v_6] = \{v_1, v_2, v_4, v_6\}$

$N[v_7] = \{v_2, v_6, v_7, v_8\}$ $\qquad N[v_8] = \{v_3, v_5, v_7, v_8\}$



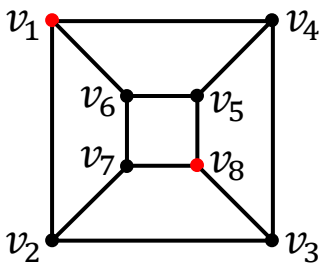
PDF (Perfect Dominating Function)

Let $f : V \to \{0, 1\}$ is a function
that maps from vertex set V to set $\{0, 1\}$ for graph $G = (V, E)$.
f is called a PDF if it is satisfying the following condition

$$\forall v \in V, \ \sum_{u \in N[v]} f(v) = 1$$

$$f(v) = x_v = \begin{cases} 1, if\ v \in A \\ 0, otherwise \end{cases}$$

assign each vertex of the PDS to 1
and all others to 0, then f is a PDF



0

$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$

$x_{v_1} = 1, \quad x_{v_2} = 0, \quad x_{v_3} = 0, \quad x_{v_4} = 0,$

$\sum_{u \in N[v_1]} x_u = x_{v_1} + x_{v_2} + x_{v_4} + x_{v_6} = 1$

$\sum_{u \in N[v_3]} x_u = x_{v_2} + x_{v_6} + x_{v_4} + x_{v_8} = 1$

$\sum_{u \in N[v_5]} x_u = x_{v_4} + x_{v_5} + x_{v_6} + x_{v_8} = 1$

$\sum_{u \in N[v_7]} x_u = x_{v_2} + x_{v_6} + x_{v_7} + x_{v_8} = 1$

1

$A = \{v_1, v_8\}$

$x_{v_5} = 0, \quad x_{v_6} = 0, \quad x_{v_7} = 0, \quad x_{v_8} = 1$

$\sum_{u \in N[v_2]} x_u = x_{v_1} + x_{v_2} + x_{v_3} + x_{v_7} = 1$

$\sum_{u \in N[v_4]} x_u = x_{v_1} + x_{v_3} + x_{v_4} + x_{v_5} = 1$

$\sum_{u \in N[v_6]} x_u = x_{v_1} + x_{v_2} + x_{v_4} + x_{v_6} = 1$

$\sum_{u \in N[v_8]} x_u = x_{v_3} + x_{v_5} + x_{v_7} + x_{v_8} = 1$

Perfect Code Cryptosystems

This cryptosystem relies on the problem of finding a PDS in the graph, which is expected to be NP-hard

public key : 3-regular graph having a PDS / secret key : PDF

1. Bob generates and publishes public key graph
2. Alice generates a polynomial ciphertext with the vertices of the graph
3. Bob gets message by substituting the variables of the vertices into a PDF value

Alice

public parameter
maximum degree of polynomial k, modulo p

Bob

2. Encryption

i. Generate ordered pairs (I_S, c_S) based on the message m

$I = \{I_1, I_2, \dots, I_t\}, I \subseteq P^*(V),\ |I_S| \le k$

$m = \sum_{S \in I} c_S \pmod p$

ii. Generate polynomial ciphertext ct

$ct(x_{v_1}, x_{v_2}, \cdots, x_{v_n}) = \sum_{S \in I} c_S \prod_{u \in S} \sum_{v \in N[u]} x_v$

pk

1. Keygen

public key : $pk \leftarrow G = (V, E),\ n = |V|$

secret key : PDF $f(v)$

$$sk \leftarrow f(v) = x_v = \begin{cases} 1, if\ v \in PDS \\ 0, otherwise \end{cases}$$

ct

3. Decryption

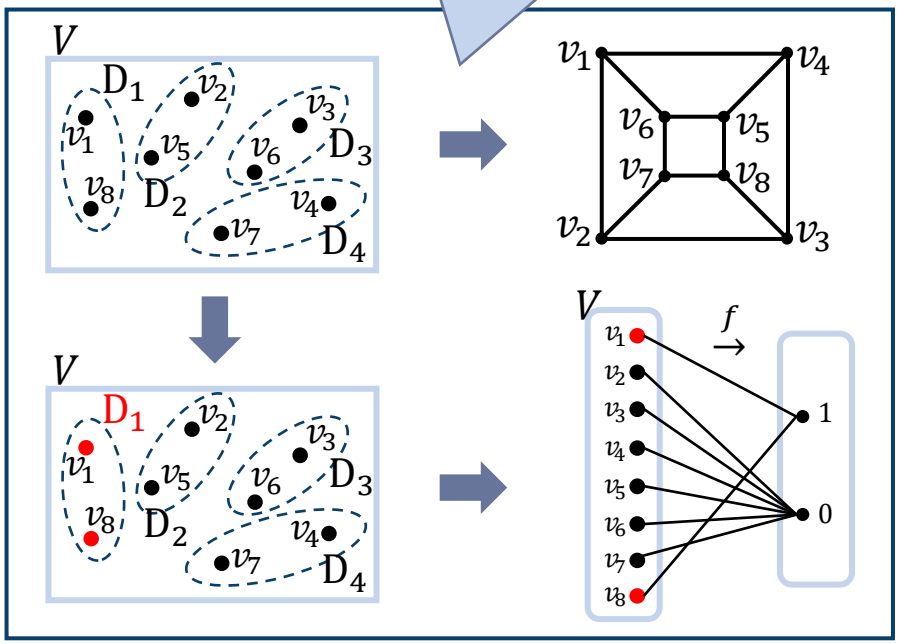
$m = ct(f(v_1), f(v_2), \dots, f(v_8))$

Example of Perfect Code Cryptosystems

public parameter
maximum degree of polynomial $k = 2$, modulo $p = 11$

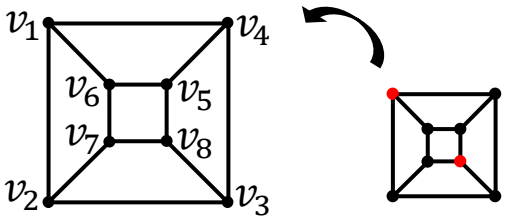
Bob

To generate a graph with PDS,
assign a one-to-one correspondence
between the vertices of each subset



1. Key generation

pk : 3-regular graph having a PDS
(PDS does not appear in published graph)



sk : PDF that depends on the PDS

$$f(v) = x_v = \begin{cases} 1, if\ v \in \{v_1, v_8\} \\ 0, otherwise \end{cases}$$

Example of Perfect Code Cryptosystems

Alice

public parameter
maximum degree of polynomial $k = 2$, modulo $p = 11$

2. Encryption

(1) Generate ordered pairs (I_S, c_S) for $m = 10$

Select a set $\mathrm{I} \subseteq P^*(V)$ $\forall \mathrm{I}_S \in \mathrm{I}$, $|\mathrm{I}_S| \leq k$

$$\mathrm{I} = \{\{v_1\}, \{v_7\}, \{v_2, v_8\}\} \xleftarrow{\$} P^*(V)$$

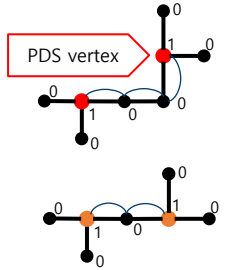
Assign an arbitrary value c corresponding to each element of I to satisfy the following condition

$$m = \sum_{\mathrm{I}_S \in \mathrm{I}} c_{\mathrm{I}_S} \pmod p$$

$$c_{\{v_1\}} = 5,\ c_{\{v_7\}} = 7,\ c_{\{v_2, v_8\}} = 9, \quad \sum c \bmod 11 = 10$$

1) A value corresponding to a vertex with a distance of 1 or 2 from the vertex belonging to the PDS = 0

▶ $x_{v_i} x_{v_j} = 0 \quad if \quad 1 \leq |v_i v_j| \leq 2$



2) PDF : $f(v) = x_v = \begin{cases} 1, if\ v \in \text{PDS} \\ 0, otherwise \end{cases}$

▶ $x_{v_i}^2 = \begin{cases} 1, if\ v \in \text{PDS} \\ 0, otherwise \end{cases}$

(2) Generate polynomial ciphertext ct

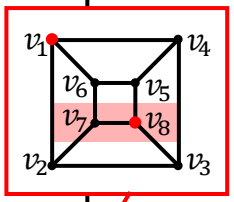
Delete terms if the distance of the multiplied vertex variables is 1 or 2,
Replace the higher order by the first order

$ct = \sum_{i=1}^{3} c_i \prod_{u \in \mathrm{I}_i} \sum_{v \in N[u]} x_v$ — invariant polynomial

$$= 5(x_{v_1} + x_{v_2} + x_{v_4} + x_{v_6}) + 7(x_{v_2} + x_{v_6} + x_{v_7} + x_{v_8}) + 9(x_{v_1} + x_{v_2} + x_{v_3} + x_{v_7})(x_{v_3} + x_{v_5} + x_{v_7} + x_{v_8})$$

$$= 5x_{v_1} + 12x_{v_2} + 5x_{v_4} + 12x_{v_6} + 7x_{v_7} + 7x_{v_8} + 9x_{v_1}x_{v_3} + 9x_{v_1}x_{v_5} + 9x_{v_1}x_{v_7} + 9x_{v_1}x_{v_8} + 9x_{v_2}x_{v_3} + 9x_{v_2}x_{v_5}$$
$$+ 9x_{v_2}x_{v_7} + 9x_{v_2}x_{v_8} + 9x_{v_3}x_{v_3} + 9x_{v_3}x_{v_5} + 9x_{v_3}x_{v_7} + 9x_{v_3}x_{v_8} + 9x_{v_7}x_{v_3} + 9x_{v_7}x_{v_5} + 9x_{v_7}x_{v_7} + 9x_{v_7}x_{v_8}$$

$$= 5x_{v_1} + 12x_{v_2} + 9x_{v_3} + 5x_{v_4} + 12x_{v_6} + 16x_{v_7} + 7x_{v_8} + 9x_{v_1}x_{v_8} + 9x_{v_2}x_{v_5}$$



$|v_7 v_8| = 1 \leq 2 \rightarrow$ delete

Example of Perfect Code Cryptosystems

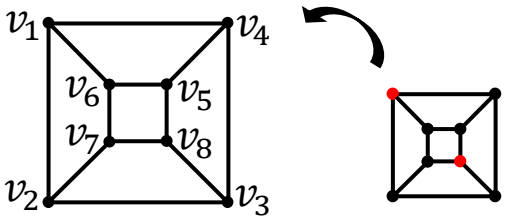
Alice

public parameter
maximum degree of polynomial $k = 2$, modulo $p = 11$

Bob

1. Key generation

pk : 3-regular graph having a PDS
(PDS does not appear in published graph)



sk : PDF that depends on the PDS

$$f(v) = x_v = \begin{cases} 1, if\ v \in \{v_1, v_8\} \\ 0, otherwise \end{cases}$$

pk

2. Encryption

(1) Generate ordered pairs (I_S, c_S) for $m = 10$

Select a set $I \subseteq P^*(V)$ $\forall I_S \in I,\ |I_S| \leq k$

$$I = \{\{v_1\}, \{v_7\}, \{v_2, v_8\}\} \xleftarrow{\$} P^*(V)$$

Assign arbitrary value c corresponding to each element of I to satisfy the following condition

$$m = \sum_{I_S \in I} c_{I_S}\ (mod\ p)$$

$c_{\{v_1\}} = 5,\ c_{\{v_7\}} = 7,\ c_{\{v_2, v_8\}} = 9$

$$\sum c = 5 + 7 + 9\ mod\ 11 = 10$$

(2) Generate polynomial ciphertext ct

$$ct = \sum_{i=1}^{3} c_i \prod_{u \in I_i} \sum_{v \in N[u]} x_v$$

$$= 5(x_{v_1} + x_{v_2} + x_{v_4} + x_{v_6}) + 7(x_{v_2} + x_{v_6} + x_{v_7} + x_{v_8})$$
$$+9(x_{v_1} + x_{v_2} + x_{v_3} + x_{v_7})(x_{v_3} + x_{v_5} + x_{v_7} + x_{v_8})$$
$$= 5x_{v_1} + 12x_{v_2} + 9x_{v_3} + 5x_{v_4} + 12x_{v_6}$$
$$+16x_{v_7} + 7x_{v_8} + 9x_{v_1}x_{v_8} + 9x_{v_2}x_{v_5}$$

ct

3. Decryption (PDF substitution)

$$ct(f(v_1), f(v_2), \ldots, f(v_n))$$
$$= 5f(v_1) + 12f(v_2) + 9f(v_3) + 5f(v_4) + 12f(v_6)$$
$$+16f(v_7) + 7f(v_8) + 9f(v_1)f(v_8) + 9f(v_2)f(v_5)$$
$$= 5 + 0 + 0 + 0 + 0 + 0 + 7 + 9 + 0 \bmod 11 = 10$$

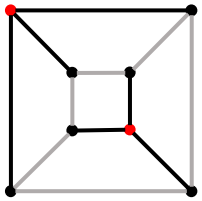
The main attack techniques that make the legacy cryptosystem difficult to use are key recovery attack and plaintext attack

Crucially, in order to defend against plaintext recovery attacks,
the size of the public key and the size of the ciphertext become very large,
resulting in very slow encryption.

Key recovery attack

The goal is to find the secret key corresponding to the public key
When attempting exhaustive key search to find a PDS of the graph,

complexity is $O\left(\binom{4n_0}{n_0}\right)$



If the size of PDS is n_0,

the number of vertices of graph is $4n_0$

Complexity of key recovery attack according to graph size

n	80	120	160	200
complexity	2^{61}	2^{93}	2^{126}	2^{158}

➡ Considering only key recovery attack, the legacy cryptosystem is not bad to use

Plaintext recovery attack

The goal is not to find PDS, but to recover the plaintext associated with the ciphertext
Generate ct' containing the number of all cases when the (I_s, c_s) pair is not known, and then apply Gauss-Jordan elimination to $ct' = ct$ to find $m' = m$

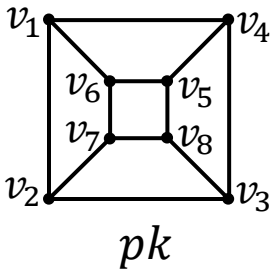
Attack algorithm

1. Select a set $\mathrm{I}' \subseteq P^*(V)$ that satisfies $\forall \mathrm{I}_S{}' \in \mathrm{I}',\ |\mathrm{I}_S{}'| \leq k,\ |\mathrm{I}'| = \sum_{j=1}^{k} \binom{n}{j}$
2. Set the integer c' corresponding to each element set of I' as an unknown value
$$c'_1,\ c'_2,\ \dots,\ c'_{|\mathrm{I}'|}$$
3. Proceed with encryption algorithm for I'
$$ct' = \sum_{i=1}^{|\mathrm{I}'|} c'_i \prod_{u \in I'_i} \sum_{v \in N[u]} x_v$$
4. Generate a system of linear equations for the $c'_1,\ c'_2,\ \dots,\ c'_{|\mathrm{I}'|}$ where $ct' = ct$
5. Calculate the RREF(reduced row echelon form) matrix by applying the Gauss-Jordan elimination to the generated system of linear equations
6. Let $\mathrm{R} = \{\mathrm{r}_1, \mathrm{r}_2, \dots, \mathrm{r}_{|\mathrm{I}'|}, \mathrm{r}_{|\mathrm{I}'|+1}\}$ be a vector obtained by adding the column vectors of the RREF matrix, then $\{\mathrm{r}_1, \mathrm{r}_2, \dots, \mathrm{r}_{|\mathrm{I}'|}, \mathrm{r}_{|\mathrm{I}'|+1}\} = \{1,\ 1,\ \dots,\ 1,\ m'\}$ in $\mathbb{Z}_p$
7. $m' = m$

Kwon, Sujin, Ju-Sung Kang, and Yongjin Yeom. "Analysis of public-key cryptography using a 3-regular graph with a perfect dominating set." 2021 IEEE Region 10 Symposium (TENSYMP). IEEE, pp. 1-6, 2021.

Example of plaintext recovery attack

Public parameters k=1, p=11 and public key graph is pk



pk

Alice's message m=5

Alice select a set $\mathrm{I} = \{\{v_1\}, \{v_6\}, \{v_7\}\}$

Arbitrary values c_S for I are $c_{\{v_1\}} = 7$, $c_{\{v_6\}} = 3$, $c_{\{v_7\}} = 6$

Ciphertext generated by Alice : $ct = 10x_{v_1} + 2x_{v_2} + 7x_{v_4} + 3x_{v_5} + 5x_{v_6} + 9x_{v_7} + 6x_{v_8}$

$$ct = 7(x_{v_1} + x_{v_2} + x_{v_4} + x_{v_6}) + 3(x_{v_1} + x_{v_5} + x_{v_6} + x_{v_7}) + 6(x_{v_2} + x_{v_6} + x_{v_7} + x_{v_8})$$

Attack process (eavesdropper got the pk and ct)

1. Construct the set I′ to include all possible cases (For $k = 1$)

$$\mathrm{I}' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$$

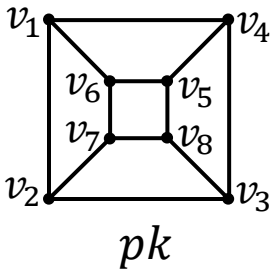
2. Set the unknown value for each element of I′ as c'_j $(1 \le j \le |\mathrm{I}'|)$

$$(\{v_1\}, c'_1), (\{v_2\}, c'_2), (\{v_3\}, c'_3), (\{v_4\}, c'_4), (\{v_5\}, c'_5), (\{v_6\}, c'_6), (\{v_7\}, c'_7), (\{v_8\}, c'_8)$$

Example of plaintext recovery attack

Public parameters k=1, p=11 and public key graph is pk

$ct = 10x_{v_1} + 2x_{v_2} + 7x_{v_4} + 3x_{v_5} + 5x_{v_6} + 9x_{v_7} + 6x_{v_8}$



pk

Attack process

3. Generate arbitrary ciphertext ct' to use in the attack

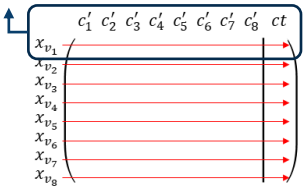
$$ct' = c_1'(x_{v_1} + x_{v_2} + x_{v_4} + x_{v_6}) + c_2'(x_{v_1} + x_{v_2} + x_{v_3} + x_{v_7}) + c_3'(x_{v_2} + x_{v_3} + x_{v_4} + x_{v_8})$$
$$+c_4'(x_{v_1} + x_{v_3} + x_{v_4} + x_{v_5}) + c_5'(x_{v_4} + x_{v_5} + x_{v_6} + x_{v_8}) + c_6'(x_{v_1} + x_{v_5} + x_{v_6} + x_{v_7})$$
$$+c_7'(x_{v_2} + x_{v_6} + x_{v_7} + x_{v_8}) + c_8'(x_{v_3} + x_{v_5} + x_{v_7} + x_{v_8})$$
$$= x_{v_1}(c_1' + c_2' + c_4' + c_6') + x_{v_2}(c_1' + c_2' + c_3' + c_7') + x_{v_3}(c_2' + c_3' + c_4' + c_8')$$
$$+x_{v_4}(c_1' + c_3' + c_4' + c_5') + x_{v_5}(c_4' + c_5' + c_6' + c_8') + x_{v_6}(c_1' + c_5' + c_6' + c_7')$$
$$+x_{v_7}(c_2' + c_6' + c_7' + c_8') + x_{v_8}(c_3' + c_5' + c_7' + c_8')$$

Compare the coefficients of each term of ct and ct' to form a system of linear equations

RASE KMU

Example of plaintext recovery attack

$$x_{v_1}(1\times c'_1+1\times c'_2+0\times c'_3+1\times c'_4+0\times c'_5+1\times c'_6+0\times c'_7+0\times c'_8)=10x_{v_1}$$



Compare the coefficients of each term of ct and ct' and form a system of linear equations

Attack process

4. Calculate the RREF matrix by applying the Gauss-Jordan elimination to the $ct' = ct$

$$\begin{array}{c|cccccccc|c} & c'_1 & c'_2 & c'_3 & c'_4 & c'_5 & c'_6 & c'_7 & c'_8 & \\ \hline & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 10 \\ & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 2 \\ & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 7 \\ & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 3 \\ & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 5 \\ & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 9 \\ & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 6 \end{array} \Rightarrow \begin{array}{c|cccccccc|c} & c'_1 & c'_2 & c'_3 & c'_4 & c'_5 & c'_6 & c'_7 & c'_8 & \\ \hline & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 7 \\ & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & -2 \\ & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & -3 \\ & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 5 \\ & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & -2 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

5. The result of adding each column vector of the RREF matrix on $\mathbb{Z}_{11}$ is:

$$\begin{array}{c|cccccccc|c} & c'_1 & c'_2 & c'_3 & c'_4 & c'_5 & c'_6 & c'_7 & c'_8 & \\ \hline & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 7 \\ & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & -2 \\ & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & -3 \\ & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 5 \\ & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & -2 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \textcircled{5} \end{array}$$

The value of each c' is not uniquely determined, but message $m = 5$ is recoverable

Plaintext recovery attack

Generate ct' containing the number of all cases when the (I_s, c_s) pair is not known, and then apply Gauss-Jordan elimination to $ct' = ct$ to find $m' = m$
Complexity is $O(t^3)$
when $t = \sum_{i=1}^{k} \binom{n}{i}$ is the number of terms in the attacker's arbitrary ciphertext

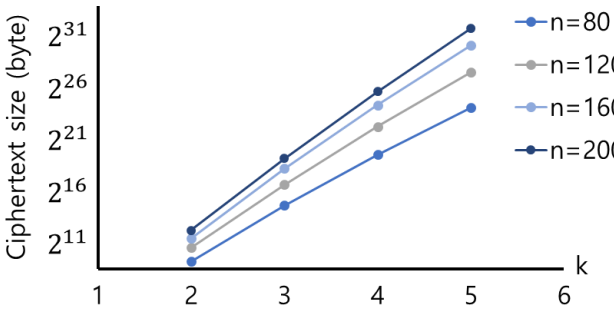
Alice selects a set I in $P^*(V)$

$$ct(x_{v_1}, x_{v_2}, \cdots, x_n) = \sum_{S \in \mathrm{I}} c_S \prod_{u \in S} \sum_{v \in N[u]} x_v$$

I′ consists of every element S of $P^*(V)$ such that $|S| \le k$

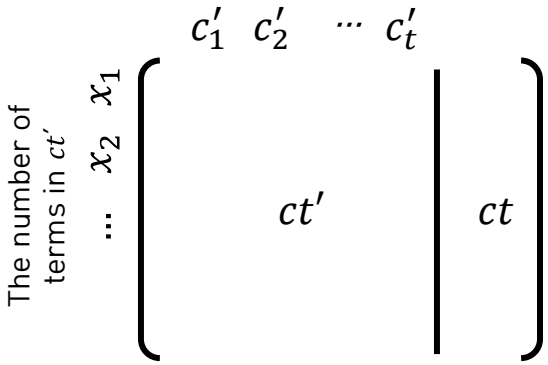
$$ct'(x_{v_1}, x_{v_2}, \cdots, x_n) = \sum_{S \in \mathrm{I}'} c'_S \prod_{u \in S} \sum_{v \in N[u]} x_v$$

The complexity of plaintext recovery attack depends on the number of polynomial terms, and it depends on the security parameters n and k



The number of terms required by n and k for the attack

▶ $\sum c_S' \prod \sum x_v = \sum c_S \prod \sum x_v$



Complexity has been drastically reduced by plaintext recovery attack

Complexity by graph size and attack technique

n	80	120	160	200
Key recovery attack	2^{61}	2^{93}	2^{126}	2^{158}
Plaintext recovery attack (k=4)	2^{62}	2^{69}	2^{74}	2^{77}

However, increasing n and k to defend against plaintext recovery attack makes it difficult for users to use the cryptosystem

performance of legacy cryptosystem (k=4)

|I|	the number of terms	running time		
		key gen	encryption	decryption
100	6,272	0.020ms	201ms	0.110ms
500	23,111	0.019ms	2260ms	0.536ms
1000	41,131	0.020ms	7364ms	0.993ms

Kwon, Sujin, Ju-Sung Kang, and Yongjin Yeom. "Analysis of public-key cryptography using a 3-regular graph with a perfect dominating set." 2021 IEEE Region 10 Symposium (TENSYMP). IEEE, pp. 1-6, 2021.

We want to increase the security against to plaintext recovery attacks without increasing n and k

If n does not increase, then the security of key recovery attack does not increase

➡ Improve the balance of security
against key recovery attack and plaintext recovery attack

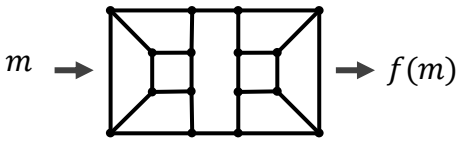
Concept of direction for improve the cryptosystem

Split a graph with the number of vertices N, maximum degree K

to l graphs with the number of vertices n_i, maximum degree k_i $(2 \leq i \leq l)$

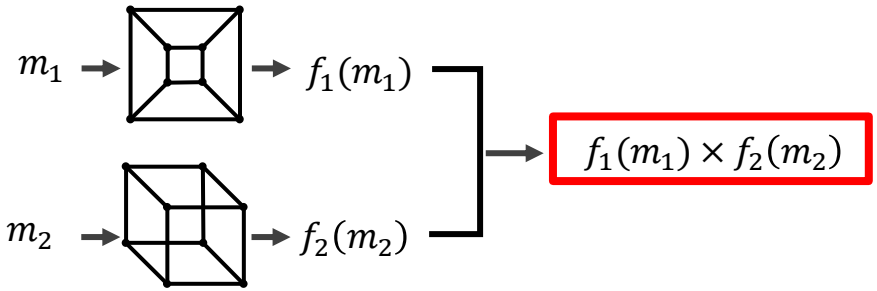
Ex

1 polynomial with maximum degree $2k$
in a graph with $2n$ vertices



It is not necessary to divide the graph into multiple graphs of the same size

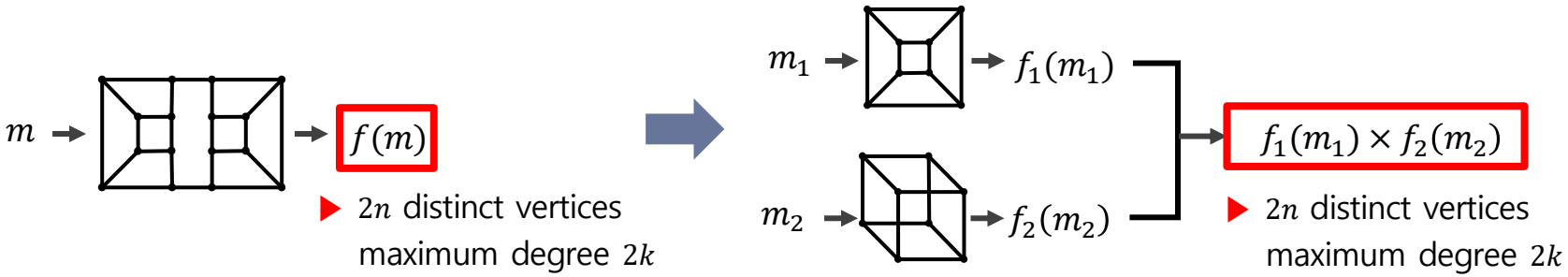
2 polynomials with maximum degree k
in two graphs with n vertices



▶ final polynomial: $2n$ distinct vertices, maximum degree $2k$

Concept of direction for improve the cryptosystem

If attacker cannot separate the polynomial into
smaller ones with lower degree generated by 2 or more graphs,
although the complexity of key recovery attack is not changed for each graphs,
the complexity of the plaintext recovery attack for ciphertext
follows the amount of operation required to generate a polynomial
with maximum degree K made by a graph consisted of N vertices



Complexity of plaintext recovery attack when n=100, k=2

one graph : 2^{36} ➡ two graph : 2^{77}

for key recovery attack when n=100 : 2^{77} Similar security level against two attacks

1st round IPCC Improved Perfect Code Cryptosystems Algorithm

Alice

public parameter
maximum degree of polynomial k, modulo p

Bob

2. Encryption

(1) Select F to combine polynomials such that the maximum order is k

$$F_{\mathcal{G}}^{k} \xleftarrow{\$} \mathcal{F}^{k}$$

(2) For F, distribute the message m to satisfy the following condition

$$m = F(m_1, m_2, \dots)$$

(3) For each m_i, generate a polynomial $f_{G_i}^{k_i}(m_i)$ with maximum degree k_i on G_i
using legacy algorithm

(4) Return ct by combining the polynomial f_i according to the form of F

$$ct \leftarrow F_{\mathcal{G}}^{k}\left(x_{v_1}, x_{v_2}, \dots\right) = F(f_1(m_1), f_2(m_2), \dots)$$

pk

Generate low-degree polynomials with multiple graphs

ct

1. Key gen

Public key : 2 or more small graphs

$$pk \leftarrow \mathcal{G} = \{G_1, G_2, \dots\}$$

Secret key : PDF using PDS of each graph
(as same as the legacy cryptosystem)

$$sk \leftarrow PDF(v) = x_v$$

$$= \begin{cases} 1, if\ v \in PDS_i \\ 0, otherwise \end{cases}$$

3. Decryption

PDF substitution
(as same as the legacy cryptosystem)

$$\mathrm{m} \leftarrow ct(PDF(v_1), PDF(v_2), \dots)$$

Select F in the family of multivariate mixing functions $\mathcal{F}^k$ with maximum degree k

Example of 1st round IPCC Algorithm

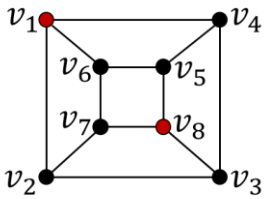
public parameter
maximum degree of polynomial k=2, modulo $p = 11$

Bob

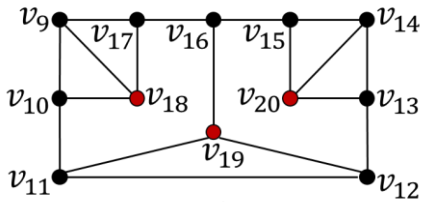
pk

1. Key gen

pk : generate 2 or more graphs



Graph G_1



Graph G_2

sk : generate PDF

as same as the legacy cryptosystem

$$PDF(v) = x_v = \begin{cases} 1, if\ v \in \{v_1, v_8, v_{18}, v_{19}, v_{20}\} \\ 0, otherwise \end{cases}$$

Alice

public parameter
maximum degree of polynomial k=2, modulo $p = 11$

2. Encryption m=4

(1) Select $F = f_1f_2 + f_3f_4$, each polynomial f_i of degree 1

$$F_G^2 \xleftarrow{\$} \mathcal{F}^2 = \{f_1f_2,\ f_1f_2+f_3,\ \ldots\}$$

The process of generating a ciphertext by low-degree polynomials over small graphs consisted of few vertices

The randomly selected F is not communicated by Alice to Bob.

(2) Distribute message m to m_1, m_2, m_3, m_4 for F

$$m_1 = 3,\ m_2 = 4, m_3 = 7,\ m_4 = 2$$

$$m = F(m_1, m_2, m_3, m_4) = m_1m_2 + m_3m_4 = 4\ mod\ 11$$

(3) For each m_i, generate a polynomial $f_i(m_i)$ on $G_i = X$ or Y using the legacy algorithm

$f_1(G_1, m_1)$
$m_1 = 3, \mathrm{I} = \{v_2\},\ c_{\{v_2\}} = 3$
$f_1 = c_{\{v_2\}} \prod_{u \in N_{[v_2]}} x_u = 3\,(x_{v_1} + x_{v_2} + x_{v_3} + x_{v_7})$

$f_2(G_2, m_2)$
$m_2 = 4, \mathrm{I} = \{v_{18}\},\ c_{\{v_{18}\}} = 4$
$f_2 = c_{\{v_{18}\}} \prod_{u \in N_{[v_{18}]}} x_u = 4\,(x_{v_9} + x_{v_{10}} + x_{v_{17}} + x_{v_{18}})$

$f_3(G_1, m_3)$
$m_3 = 7, \mathrm{I} = \{v_8\},\ c_{\{v_8\}} = 7$
$f_3 = c_{\{v_8\}} \prod_{u \in N_{[v_8]}} x_u = 7\,(x_{v_3} + x_{v_5} + x_{v_7} + x_{v_8})$

$f_4(G_2, m_4)$
$m_4 = 2, \mathrm{I} = \{v_{11}\},\ c_{\{v_{11}\}} = 2$
$f_4 = c_{\{v_{11}\}} \prod_{u \in N_{[v_{11}]}} x_u = 2\,(x_{v_{10}} + x_{v_{11}} + x_{v_{12}} + x_{v_{19}})$

(4) Return ct by combining the polynomial f_i according to the form of F

$$ct = f_1(G, m_1) \times f_2(G_2, m_2) + f_3(G_1\ m_3) \times f_4(G_2, m_4)$$

$$= 3(x_{v_1} + x_{v_2} + x_{v_3} + x_{v_7})4(x_{v_9} + x_{v_{10}} + x_{v_{17}} + x_{v_{18}}) + 7(x_{v_3} + x_{v_5} + x_{v_7} + x_{v_8})2(x_{v_{10}} + x_{v_{11}} + x_{v_{12}} + x_{v_{19}})$$

$$= x_{v_1}x_{v_9} + x_{v_2}x_{v_9} + x_{v_3}x_{v_9} + x_{v_7}x_{v_9} + x_{v_1}x_{v_{10}} + x_{v_2}x_{v_{10}} + 4x_{v_3}x_{v_{10}} + 4x_{v_7}x_{v_{10}} + x_{v_1}x_{v_{17}} + x_{v_2}x_{v_{17}} + x_{v_3}x_{v_{17}} + x_{v_7}x_{v_{17}} + x_{v_1}x_{v_{18}} + x_{v_2}x_{v_{18}} + x_{v_3}x_{v_{18}} + x_{v_7}x_{v_{18}}$$
$$+3x_{v_5}x_{v_{10}} + 3x_{v_8}x_{v_{10}} + 3x_{v_3}x_{v_{11}} + 3x_{v_5}x_{v_{11}} + 3x_{v_7}x_{v_{11}} + 3x_{v_8}x_{v_{11}} + 3x_{v_3}x_{v_{12}} + 3x_{v_5}x_{v_{12}} + 3x_{v_7}x_{v_{12}} + 3x_{v_8}x_{v_{12}} + 3x_{v_3}x_{v_{19}} + 3x_{v_5}x_{v_{19}} + 3x_{v_7}x_{v_{19}} + 3x_{v_8}x_{v_{19}}$$

Example of 1st round IPCC Algorithm

public parameter
maximum degree of polynomial k=2, modulo $p = 11$

$$
\begin{aligned}
ct = \; & x_{v_1}x_{v_9} + x_{v_2}x_{v_9} + x_{v_3}x_{v_9} + x_{v_7}x_{v_9} + x_{v_1}x_{v_{10}} \\
& + x_{v_2}x_{v_{10}} + 4x_{v_3}x_{v_{10}} + 4x_{v_7}x_{v_{10}} + x_{v_1}x_{v_{17}} + x_{v_2}x_{v_{17}} \\
& + x_{v_3}x_{v_{17}} + x_{v_7}x_{v_{17}} + x_{v_1}x_{v_{18}} + x_{v_2}x_{v_{18}} + x_{v_3}x_{v_{18}} \\
& + x_{v_7}x_{v_{18}} + 3x_{v_5}x_{v_{10}} + 3x_{v_8}x_{v_{10}} + 3x_{v_3}x_{v_{11}} + 3x_{v_5}x_{v_{11}} \\
& + 3x_{v_7}x_{v_{11}} + 3x_{v_8}x_{v_{11}} + 3x_{v_3}x_{v_{12}} + 3x_{v_5}x_{v_{12}} + 3x_{v_7}x_{v_{12}} \\
& + 3x_{v_8}x_{v_{12}} + 3x_{v_3}x_{v_{19}} + 3x_{v_5}x_{v_{19}} + 3x_{v_7}x_{v_{19}} + 3x_{v_8}x_{v_{19}}
\end{aligned}
$$

ct →

Bob

3. Decryption (PDF substitution)

$$sk = PDF(v)$$

$$= x_v = \begin{cases} 1, if\ v \in \{v_1, v_8, v_{18}, v_{19}, v_{20}\} \\ 0, otherwise \end{cases}$$

$$m \leftarrow ct(PDF(v_1), PDF(v_2), \dots)$$

$$
\begin{aligned}
ct(PDF) = & \; 0 + 0 + 0 + 0 + 0 \\
& + 0 + 0 + 0 + 0 + 0 \\
& + 0 + 0 + 1 + 0 + 0 \\
& + 0 + 0 + 0 + 0 + 0 \\
& + 0 + 0 + 0 + 0 + 3 \\
= & \; 4 \bmod 11
\end{aligned}
$$

An attacker cannot factor ct into a short polynomials that following the form F

▶ When applying the plaintext recovery attack, it cannot be perform for each f_i, but it must be performed on a higher degree polynomial F

3. Round 1 version of IPCC

The main improvement is the inclusion of a process for Alice to choose F,
and Bob does not need any information about this
because he just needs to substitute the PDF value for variables to get the original message

➡ F is secret to everyone except Alice

This approach prevents attackers from using plaintext recovery attacks on each f_i
since they only have access to the ciphertext ct

Even if the example can be factored and each polynomial can be attacked,
the real cryptosystem uses a ciphertext with many more vertices and high degree, amaking it more resistant to attacks

To solve ct, an attacker would have to do a lot more computation than Bob

- attacker : solve F (generate and check all possible terms)
- encryptor : generate each f_i forming a ciphertext in the form of F
  (amount of computation in the process of combining is negligible)

➡ We expect that users encrypting messages will gain significant advantages
over attackers in terms of speed and memory

Improving performance of the implemented algorithms

performance comparison ($n = 200, k = 4$)

	key size		number of terms	keygen time	enc time	dec time
PDF cryptosystem	pk	$4800-\text{byte}$	2.4×10^4	1ms	84,781ms	4ms
	sk	$400-\text{byte}$				
IPCC (1 graph)	pk	$2400-\text{byte}$	2.6×10^4	0.03ms	2.41ms	3.36ms
	sk	$200-\text{byte}$				
IPCC (2 graphs)	pk	$4800-\text{byte}$	2.3×10^4	1.06ms	0.35ms	0.33ms
	sk	$400-\text{byte}$				

New attack technique and problem analysis for 1st round IPCC

Paragraphs about new attack technique among the feedback received on the proposed algorithm

> As an example, consider the first example in the KAT for the case of f1. This is given a message m = 18790. The ciphertext produced by the reference implementation contains the following list of coefficients (here stated without their multiplicities): [35, 9087, 14460, 16002, 16620, 21637, 22560, 24760, 33530, 36038, 36868, 38564, 39587, 39792, 62376].
> Summing these up gives us 411916 = 18790 mod 65521, which indeed is the plaintext. Note that the KAT file shows the hash of the ciphertext, not the ciphertext itself. We ran this attack on ciphertexts produced by the KAT. There are some few cases (2 out of 100 for f1, 0 out of 100 for f3, 8 out of 100 for f4) where this simple attack does not give the plaintext: in these cases, there are more than 15 coefficients, because variables repeated leading to combinations. We are still working on tracing through those to determine which of the coefficients we need to skip in summing up. We think that counting the frequency of occurrence will give us information. But we wanted to announce our findings so far as a fast attack with a success probability of more than 90% means that the system is typically broken.

To increase the attack complexity, the degree k must be large,

but the coefficients of high degree terms do not mix

Each coefficient generated in the encryption process is shared by 4^k terms

These two reasons appear to be the root of the problem

➡ The fundamental problem of the encryption process in the previous version

IPCC algorithm strengthening the encryption process

Alice

public parameter
maximum degree of polynomial k, modulo p

Bob 

2. Encryption

(1) Select element $F_{\mathcal{G}} \overset{\$}{\leftarrow} \mathcal{F}$

(2) For F, distribute the message m

$$m = F(m_1, m_2, \dots)$$

(3) For each m_i, generate a polynomial $ct_i(m_i)$ with maximum degree k_i on G_i using the previous algorithm

(3-1) Distribute the m_i to $\tilde{m}_1, \tilde{m}_2, \dots$ to satisfy $\sum \tilde{m}_j = m_i$

(3-1-1) For $\tilde{m}_j$, choose a vertex u_j

(3-1-2) Select invariant polynomial $\tilde{f}_l$ of degree $\leq k_i - 1$ not including x_{u_j} $(0 \leq l \leq 3)$

(3-1-3) Calculate constant a_l to satisfy $\tilde{m}_j = \tilde{f}_l - a_l$

(3-1-4) Generate $\tilde{ct}_j = \sum(\tilde{f}_l - a_l)\, x_{u_l}$ for $\tilde{m}_j$ $(u_l \in N[u_j])$

(3-2) For m_i, pre-ciphertext is $ct_i = \sum \tilde{ct}_j$

(4) Combine the polynomials f_i according to the form F

$$ct \leftarrow F_{\mathcal{G}}^{k}(x_{v_1}, x_{v_2}, \dots) = F(ct_1(m_1), ct_2(m_2), \dots)$$

(5) Genertate ct reduced by all graphs of $\mathcal{G}$

pk

ct

1. Key gen

Public key : 2 or more small graphs that share a set of vertices and PDS

$$pk \leftarrow \mathcal{G} = \{G_1, G_2, \dots\}$$

Secret key : PDF using PDS

$$sk \leftarrow PDF(v)$$

$$= x_v = \begin{cases} 1, if\ v \in PDS \\ 0, otherwise \end{cases}$$

3. Decryption

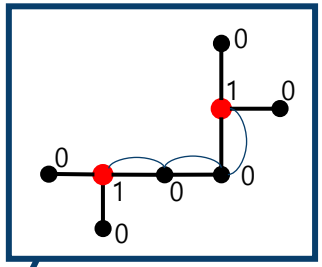
PDF substitution
(as same as the previous cryptosystem)

$$\mathrm{m} \leftarrow ct(PDF(v_1), PDF(v_2), \dots)$$

Concepts of Invariant polynomial in the new algorithm

The invariant polynomial of degree k is consisted according to

$$\sum(\tilde{f}_l - a_l)\, x_{u_l} \qquad \text{for } \tilde{m}_j \ (u_l \in N[u_j])$$



Since $l = 0, 1, 2 \text{ or } 3$ on 3-regular graph,

invariant polynomial will look like as $(\tilde{f}_0 - a_0)x_{u_0} + (\tilde{f}_1 - a_1)x_{u_1} + (\tilde{f}_2 - a_2)x_{u_2} + (\tilde{f}_3 - a_3)x_{u_3}$

and $\tilde{f}_l$ is invariant polynomial of degree $\leq k - 1$

Due to the property PDF, only the coefficient $(\tilde{f}_X - a_X)$ for x_{u_X} s.t. $x_{u_X} = 1$ where $u_X \in N[u_j]$ is meaningful as a valid value

In the previous algorithm,

all $\tilde{f}_l - a_l$ were identical to the invariant polynomial for x_{u_l}

(ex. $\left(x_{u_0'} + x_{u_1'} + x_{u_2'} + x_{u_3'}\right)x_{u_0} + \left(x_{u_0'} + x_{u_1'} + x_{u_2'} + x_{u_3'}\right)x_{u_1} + \left(x_{u_0'} + x_{u_1'} + x_{u_2'} + x_{u_3'}\right)x_{u_2} + \left(x_{u_0'} + x_{u_1'} + x_{u_2'} + x_{u_3'}\right)x_{u_3}$
$= \left(x_{u_0'} + x_{u_1'} + x_{u_2'} + x_{u_3'}\right)(x_{u_0} + x_{u_1} + x_{u_2} + x_{u_3})$)

Example

Bob 

public parameter
maximum degree of polynomial $k = 2$, modulo $p = 13$

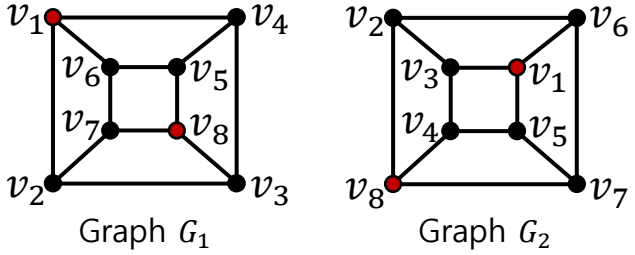
1. Key gen

Public key : 2 graphs that share a set of vertices and PDS

$$pk \leftarrow \mathcal{G} = \{G_1, G_2\}$$

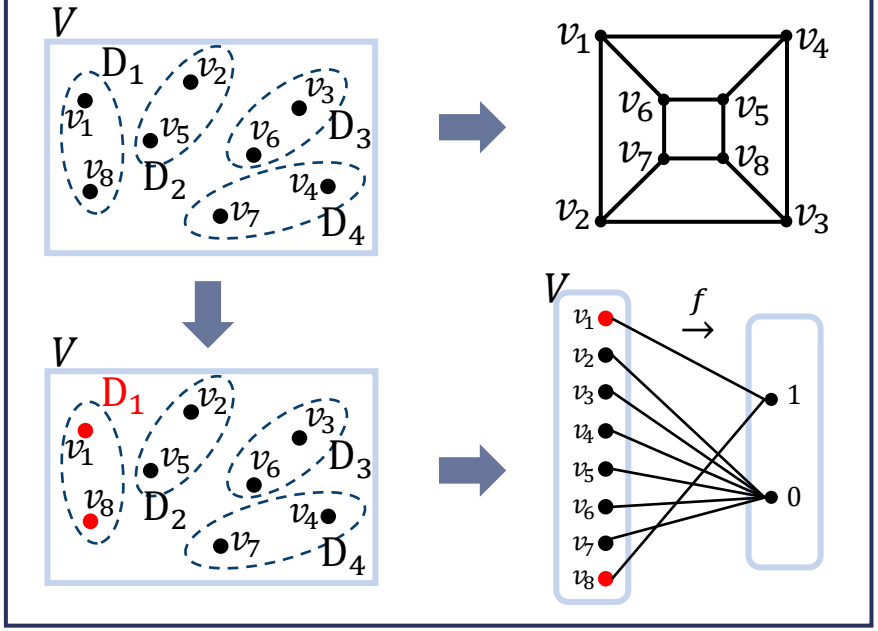
Secret key : PDF using PDS

$$sk \leftarrow PDF(v) = x_v = \begin{cases} 1, if\ v \in PDS \\ 0, otherwise \end{cases}$$

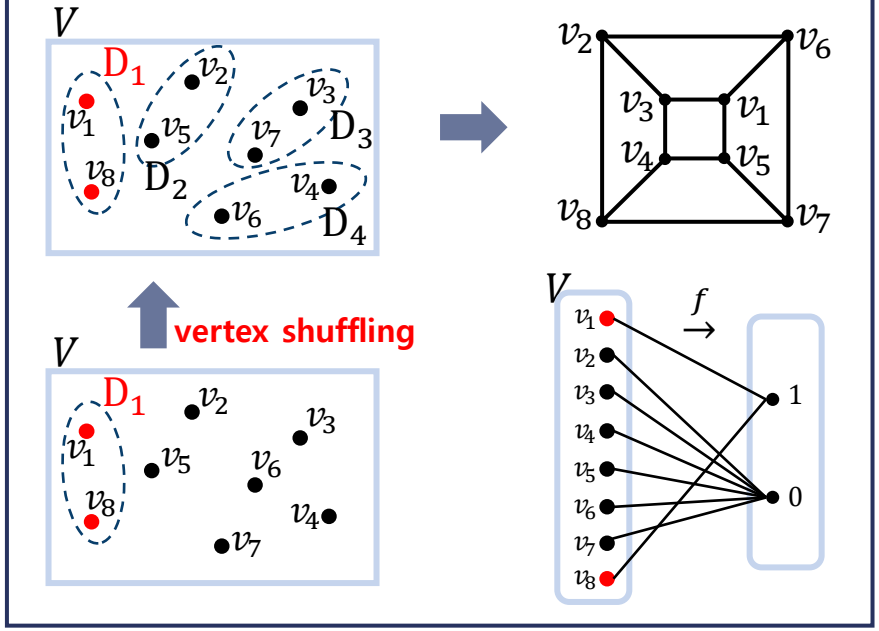


$$sk \leftarrow PDF(v) = \begin{cases} 1, if\ v \in \{v_1, v_8\} \\ 0, otherwise \end{cases}$$

Graph G_1



Graph G_2



Example

Bob 

public parameter
maximum degree of polynomial $k = 2$, modulo $p = 13$

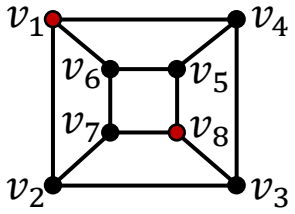
1. Key gen

Public key : 2 graphs that share a set of vertices and PDS

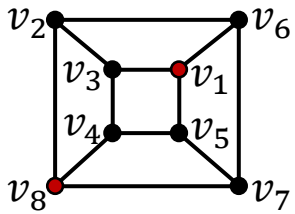
$$pk \leftarrow \mathcal{G} = \{G_1, G_2\}$$

Secret key : PDF using PDS

$$sk \leftarrow PDF(v) = x_v = \begin{cases} 1, if\ v \in PDS \\ 0, otherwise \end{cases}$$



Graph G_1



Graph G_2

$$sk \leftarrow PDF(v) = \begin{cases} 1, if\ v \in \{v_1, v_8\} \\ 0, otherwise \end{cases}$$

Public key graph set $\mathcal{G}$

Graph G_1

$$N[v_1] = \{v_1, v_2, v_4, v_6\}$$
$$N[v_2] = \{v_1, v_2, v_3, v_7\}$$
$$N[v_3] = \{v_2, v_3, v_4, v_8\}$$
$$N[v_4] = \{v_1, v_3, v_4, v_5\}$$
$$N[v_5] = \{v_4, v_5, v_6, v_8\}$$
$$N[v_6] = \{v_1, v_5, v_6, v_7\}$$
$$N[v_7] = \{v_2, v_6, v_7, v_8\}$$
$$N[v_8] = \{v_3, v_5, v_7, v_8\}$$

Graph G_2

$$N[v_1] = \{v_1, v_3, v_5, v_6\}$$
$$N[v_2] = \{v_2, v_3, v_6, v_8\}$$
$$N[v_3] = \{v_1, v_2, v_3, v_4\}$$
$$N[v_4] = \{v_3, v_4, v_5, v_8\}$$
$$N[v_5] = \{v_1, v_4, v_5, v_7\}$$
$$N[v_6] = \{v_1, v_2, v_6, v_7\}$$
$$N[v_7] = \{v_5, v_6, v_7, v_8\}$$
$$N[v_8] = \{v_2, v_4, v_7, v_8\}$$

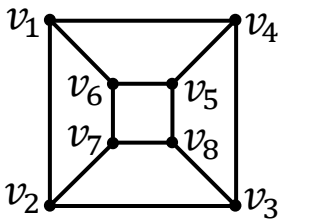
Example

Alice

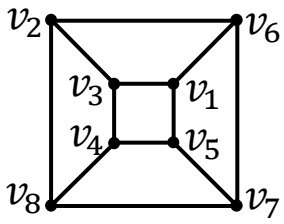
public parameter

maximum degree of polynomial $k = 2$, modulo $p = 13$

Public key graph set $\mathcal{G}$



Graph G_1



Graph G_2

2. Encryption

(1) Select element $F_{\mathcal{G}} \xleftarrow{\$} \mathcal{F}$

(2) For F, distribute the message m

$$m = F(m_1, m_2, \dots)$$

(3) For each m_i, generate a polynomial $ct_i(m_i)$ with maximum degree k_i on G_i using legacy algorithm

(3-1) Distribute the m_i to $\tilde{m}_1, \tilde{m}_2, \dots$ to satisfy $\sum \tilde{m}_j = m_i$

(3-1-1) For $\tilde{m}_j$, choose a vertex u_j

(3-1-4) Generate $\tilde{ct}_j = \sum(\tilde{f}_l - a_l)\, x_{u_l}$ for $\tilde{m}_j$ $(u_l \in N[u_j])$

(3-2) For m_i, pre-ciphertext is $ct_i = \sum \tilde{ct}_j$

(1) $F = f_1 \times f_2$

(2) $m = 12 = 3 \times 4 = m_1 \times m_2 = F(m_1, m_2)$

(3) Encrypt m_1 to polynomial ct_1 of degree 1 on G_1, m_2 to polynomial ct_2 of degree 2 on G_2

(3-1) For $m_1 = 3$, let $\tilde{m}_1 = 1, \tilde{m}_2 = 2$ $(\tilde{m}_1 + \tilde{m}_2 = m_1)$

(3-1-1) Select vertex v_2 for $\tilde{m}_1$, vertex v_3 for $\tilde{m}_2$

(3-1-4) $\tilde{ct}_1 = x_{v_1} + x_{v_2} + x_{v_3} + x_{v_7}$, $\tilde{ct}_2 = 2\left(x_{v_2} + x_{v_3} + x_{v_4} + x_{v_8}\right)$

(3-2) $ct_1 = x_{v_1} + 3x_{v_2} + 3x_{v_3} + 2x_{v_4} + x_{v_7} + 2x_{v_8}$

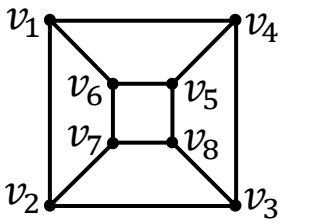
Example

Alice

public parameter

maximum degree of polynomial $k = 2$, modulo $p = 13$

Public key graph set $\mathcal{G}$



2. Encryption

(3) For each m_i, generate a polynomial $ct_i(m_i)$
with maximum degree k_i on G_i using legacy algorithm

(3-1) Distribute the m_i to $\tilde{m}_1, \tilde{m}_2, \ldots$ to satisfy $\sum \tilde{m}_j = m_i$

(3-1-1) For $\tilde{m}_j$, choose a vertex u_j

(3-1-2) Select invariant polynomial $\tilde{f}_l$ of degree $\leq k_i - 1$
not consisted by x_{u_j} $(0 \leq l \leq 3)$

(3-1-3) Calculate constant $\tilde{a}_l$ to satisfy $\tilde{m}_j = \tilde{f}_l - a_l$

(3) m_2 to polynomial ct_2 of degree $2(= k)$ on G_2

(3-1) For $m_2 = 4$, $\tilde{m} = 4$ $(\tilde{m} = m_1)$

$\tilde{ct}_j = \sum(\tilde{f}_l - a_l)\, x_{u_l}$ for $\tilde{m}_j$ $(u_l \in N[u_j])$

(3-1-1) Select vertex v_6 for $\tilde{m}$

(3-1-2) Select invariant polynomials $\tilde{f}_l$ of degree $\leq k - 1 = 1$ not including x_{v_6}

$$\tilde{f}_0 = 3\left(x_{v_1} + x_{v_2} + x_{v_3} + x_{v_4}\right),\ \tilde{f}_1 = \left(x_{v_3} + x_{v_4} + x_{v_5} + x_{v_8}\right),$$
$$\tilde{f}_2 = 5\left(x_{v_1} + x_{v_4} + x_{v_5} + x_{v_7}\right),\ \tilde{f}_3 = 4\left(x_{v_2} + x_{v_4} + x_{v_7} + x_{v_8}\right)$$

(3-1-3) Calculate constant a_l to satisfy $\tilde{m} = 4 = \tilde{f}_l - a_l$

$$a_0 = -1, a_1 = -3, a_2 = 1, a_3 = 0$$

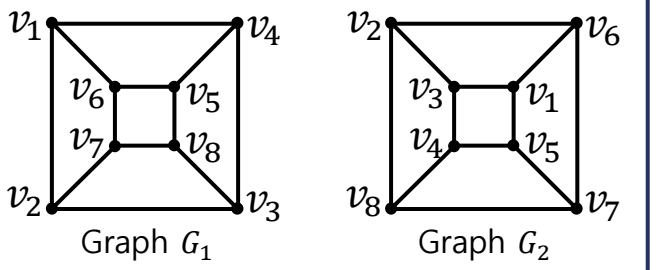
Example

Alice

public parameter

maximum degree of polynomial $k = 2$, modulo $p = 13$

Public key graph set $\mathcal{G}$



2. Encryption

(3-1-4) Generate $\tilde{ct}_j = \sum(\tilde{f}_l - a_l)\, x_{u_l}$ for $\tilde{m}_j$ $(u_l \in N[u_j])$

(3-1-5) Reduce the polynomial by neighbor relation of G_i

(3-2) For m_i, pre-ciphertext is $ct_i = \sum \tilde{ct}_j$

(4) Combine the polynomials f_i according to the form F

$$ct \leftarrow F_{\mathcal{G}}^{k}(x_{v_1}, x_{v_2}, \dots) = F(ct_1(m_1), ct_2(m_2), \dots)$$

Due to the property PDF,
only the coefficient $(\tilde{f}_1 - a_1)$ for x_{u_1}
is meaningful as a valid value

$$\text{(3-1-4) } \tilde{ct} = (\tilde{f}_0 - a_0)x_{u_0} + (\tilde{f}_1 - a_1)x_{u_1} + (\tilde{f}_2 - a_2)x_{u_2} + (\tilde{f}_3 - a_3)x_{u_3} \quad (u_l \in \{v_1, v_2, v_6, v_7\})$$

$$= \{3(x_{v_1} + x_{v_2} + x_{v_3} + x_{v_4}) + 1\}x_{v_1} + \{(x_{v_3} + x_{v_4} + x_{v_5} + x_{v_8}) + 3\}x_{v_2}$$

$$+ \{5(x_{v_1} + x_{v_4} + x_{v_5} + x_{v_7}) - 1\}x_{v_6} + 4(x_{v_2} + x_{v_4} + x_{v_7} + x_{v_8})x_{v_7}$$

$$\text{(3-1-5) } \tilde{ct} = (3x_{v_1} + 1)x_{v_1} + (x_{v_5} + 3)x_{v_2} + (5x_{v_4} + 1)x_{v_6} + (4x_{v_7})x_{v_7}$$

$$= 3x_{v_1} + x_{v_1} + x_{v_2}x_{v_5} + 3x_{v_2} + 5x_{v_4}x_{v_6} + x_{v_6} + 4x_{v_7}$$

$$= 4x_{v_1} + x_{v_2}x_{v_5} + 3x_{v_2} + 5x_{v_4}x_{v_6} + x_{v_6} + 4x_{v_7} = ct_2$$

$$\text{(4) } ct = ct_1 \times ct_2 = (x_{v_1} + 3x_{v_2} + 3x_{v_3} + 2x_{v_4} + x_{v_7} + 2x_{v_8})(4x_{v_1} + x_{v_2}x_{v_5} + 3x_{v_2} + 5x_{v_4}x_{v_6} + x_{v_6} + 4x_{v_7})$$

Example

Alice

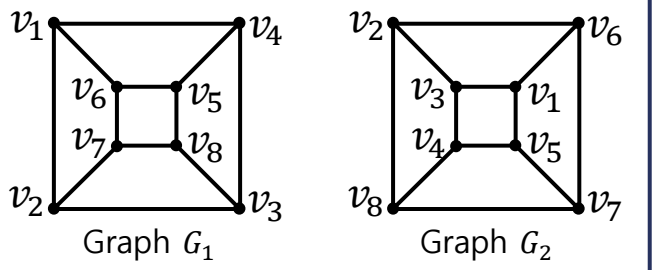
public parameter

maximum degree of polynomial $k = 2$, modulo $p = 13$

2. Encryption

(5) Generate ct reducted by all graphs of $\mathcal{G}$

Public key graph set $\mathcal{G}$



$$(5)\ ct = \left(x_{v_1} + 3x_{v_2} + 3x_{v_3} + 2x_{v_4} + x_{v_7} + 2x_{v_8}\right)4x_{v_1} + \left(x_{v_1} + 3x_{v_2} + 3x_{v_3} + 2x_{v_4} + x_{v_7} + 2x_{v_8}\right)x_{v_2}x_{v_5}$$
$$+\left(x_{v_1} + 3x_{v_2} + 3x_{v_3} + 2x_{v_4} + x_{v_7} + 2x_{v_8}\right)3x_{v_2} + \left(x_{v_1} + 3x_{v_2} + 3x_{v_3} + 2x_{v_4} + x_{v_7} + 2x_{v_8}\right)5x_{v_4}x_{v_6}$$
$$+\left(x_{v_1} + 3x_{v_2} + 3x_{v_3} + 2x_{v_4} + x_{v_7} + 2x_{v_8}\right)x_{v_6} + \left(x_{v_1} + 3x_{v_2} + 3x_{v_3} + 2x_{v_4} + x_{v_7} + 2x_{v_8}\right)4x_{v_7}$$
$$= \left(x_{v_1} + 2x_{v_8}\right)4x_{v_1} + \left(3x_{v_2}\right)x_{v_2}x_{v_5} + \left(3x_{v_2}\right)3x_{v_2} + \left(x_{v_7}\right)4x_{v_7}$$
$$= 4x_{v_1} + 8x_{v_1}x_{v_8} + 3x_{v_2}x_{v_5} + 9x_{v_2} + 4x_{v_7}$$

As the size of graph increases, the number of unreduced terms is expected to increase

Although generated by multiplying polynomial of degree 2, the maximum degree of ciphertext is 2

- Need to study how to ensure maximum order of ciphertext generated by multiplying low-degree polynomials

Example

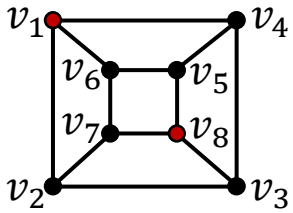
Bob 

public parameter
maximum degree of polynomial $k = 2$, modulo $p = 13$

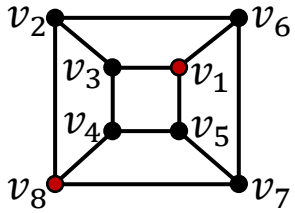
3. Decryption

Secret key : PDF using PDS

$$sk \leftarrow PDF(v) = x_v = \begin{cases} 1, if\ v \in PDS \\ 0, otherwise \end{cases}$$



Graph G_1



Graph G_2

$$ct = 4x_{v_1} + 8x_{v_1}x_{v_8} + 3x_{v_2}x_{v_5} + 9x_{v_2} + 4x_{v_7}$$

$$ct(PDF(v_1), PDF(v_2), \dots) = 4 + 8 = 12 = message$$

$$sk \leftarrow PDF(v) = \begin{cases} 1, if\ v \in \{v_1, v_8\} \\ 0, otherwise \end{cases}$$

Message cannot be obtained
only by the sum of the coefficients
(addressing vulnerability revealed by the feedback
by generating meaningless coefficients)

Future work

- Meaning of the method using multi-graph

  To an attacker, it do not seem like to make much of difference from using just one graph

- Security analysis to known attacks
- Generation of an algorithm for randomly selected F
- Optimize memory usage
- Ciphertext packing method

Improved Perfect Code Cryptosystem

감사합니다