

ШИНЖЛЭХ УХААН, ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ  
Мэдээлэл, Холбооны Технологийн Сургууль



Амгаланбаатарын Мягмарцэрэн

## Прокси дахин шифрлэх схемийн туршилтын системийг хөгжүүлэх нь

БАКАЛАВРЫН ТӨГСӨЛТИЙН АЖИЛ

Улаанбаатар хот

ШИНЖЛЭХ УХААН, ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ  
Мэдээлэл, Холбооны Технологийн Сургууль

Мэдээллийн сүлжээ, аюулгүй байдлын салбар

Прокси дахин шифрлэх схемийн  
туршилтын системийг хөгжүүлэх нь

Мэргэжлийн индекс: D061940

Мэргэжил: Мэдээллийн системийн аюулгүй байдал

Удирдагч: доктор (Ph.D) В.Нямсүрэн

Зөвлөгч: доктор (Ph.D), Ц.Энхтөр

магистр Ц.Манлайбаатар

Гүйцэтгэгч: А.Мягмарцэрэн

Улаанбаатар хот

2023 он 6 сар

Батлав. Мэдээллийн сүлжээ, аюулгүй байдлын салбарын эрхлэгч:

..... /доктор (Ph.D) Б.Мөнхбаяр/

Удирдагч: ..... /доктор (Ph.D) В.Нямсүрэн/

### ДИПЛОМЫН ТӨСӨЛ ГҮЙЦЭТГЭХ ТӨЛӨВЛӨГӨӨ

**Дипломын төслийн сэдэв:**

**Монгол:** ” Прокси дахин шифрлэх схемийн туршилтын системийг хөгжүүлэх нь”

**Англи:** ” Developing Prototype System of Proxy Re-Encryption Scheme”

**Төслийн зорилго:** Proxy Re-Encryption схемийн хэрэглээнүүдийг судалж, нэгэн хэрэглээг хэрэгжүүлэх туршилтын систем хөгжүүлэх

**Гүйцэтгэх оюутны овог нэр:**

А.Мягмарцэрэн/В190970106/

**Холбоо барих утас:**

99754252

№	Ажлын бүлэг, хэсгийн нэр	эзлэх хувь	дуусах хугацаа
Бүлэг №1. Өгөгдөл хуваалцах үйлчилгээний тухай			
1	1.1 Өгөгдөл хуваалцах үйлчилгээний тухай 1.2 Өгөгдлийн аюулгүй байдал 1.3 Шифрлэх схемүүд 1.4 Файл шифрлэх аргууд	20%	
Бүлэг №2. Прокси дахин шифрлэлтэд суурилсан файл хуваалцах систем			
2	2.1 Прокси дахин шифрлэлт 2.2 Хөгжүүлэх технологи, хэл сонгох 2.3 Хөгжүүлэлтийн орчин бэлдэх	40%	
Бүлэг №3. Прокси дахин шифрлэлтэд суурилсан файл хуваалцах систем хөгжүүлэх			
3	3.1 Системийн шаардлага 3.2 Системийн загвар 3.3 Системийн хөгжүүлэх 3.4 Файл хуваалцах системийг турших	40%	
Бүлэг №4. Ерөнхий дүгнэлт			

Төлөвлөгөөг боловсруулсан оюутан: ..... /А.Мягмарцэрэн/

## ТӨГСӨЛТИЙН АЖЛЫН ҮЗЛЭГИЙН ХУУДАС

Оюутны код: B190970106

Оюутны нэр: А.Мягмарцэрэн

Сэдвийн монгол нэр: ” Прокси дахин шифрлэх схемийн туршилтын системийг хөгжүүлэх нь”

Сэдвийн англи нэр: ” Developing Prototype System of Proxy Re-Encryption Scheme”

Удирдагч багш: доктор (Ph.D) В.Нямсүрэн

Зөвлөгч багш: доктор (Ph.D), Ц.Энхтөр, магистр Ц.Манлайбаатар

№	Үзлэгийн гүйцэтгэл	Гүйцэтгэлийн 30% -с багагүй байна.	Огноо	Удирдагч доктор (Ph.D) В.Нямсүрэн багшийн гарын үсэг
1	Үзлэг-1		IV/03-IV/07	

Багшийн товч зөвлөгөө, тайлбар:

.....

.....

.....

.....

.....

.....

.....

Үзлэг-1 хийсэн багш: ..... /доктор (Ph.D) В.Нямсүрэн/

№	Үзлэгийн гүйцэтгэл	Авсан оноо (10 оноо)	Гүйцэтгэлийн 50% -с багагүй байна.	Огноо	доктор (Ph.D), Ц.Энхтөр багшийн гарын үсэг
1	Үзлэг-2			IV/17-IV/21	

Багшийн товч зөвлөгөө, тайлбар:

.....

.....

.....

.....

.....

.....

.....

Үзлэг-2 хийсэн багш: ..... /доктор (Ph.D), Ц.Энхтөр/

## ТӨГСӨЛТИЙН АЖЛЫН ҮЗЛЭГИЙН ХУУДАС

Оюутны код: B190970106

Оюутны нэр: А.Мягмарцэрэн

Сэдвийн монгол нэр: ” Прокси дахин шифрлэх схемийн туршилтын системийг хөгжүүлэх нь”

Сэдвийн англи нэр: ” Developing Prototype System of Proxy Re-Encryption Scheme”

Удирдагч багш: доктор (Ph.D) В.Нямсүрэн

Зөвлөгч багш: доктор (Ph.D), Ц.Энхтөр, магистр Ц.Манлайбаатар

№	Үзлэгийн гүйцэтгэл	Авсан оноо (10 оноо)	Гүйцэтгэлийн 70% -с багагүй байна.	Огноо	магистр Ц.Манлайбаатар багшийн гарын үсэг
1	Үзлэг-3			V/08-V/12	

Багшийн товч зөвлөгөө, тайлбар:

.....  
.....  
.....  
.....  
.....  
.....  
.....

Үзлэг-3 хийсэн багш: ..... /магистр Ц.Манлайбаатар/

№	Үзлэгийн гүйцэтгэл	Гүйцэтгэлийн 90% -с багагүй байна.	Огноо	Удирдагч доктор (Ph.D) В.Нямсүрэн багшийн гарын үсэг
1	Үзлэг-4		V/15-V/19	

№	Удирдагч доктор (Ph.D) В.Нямсүрэн багшийн үнэлгээ (30 оноо)	Огноо	Удирдагч багшийн гарын үсэг
1		V/17	

Удирдагч багш: ..... /доктор (Ph.D) В.Нямсүрэн/

*Жич: Удирдагч багш өөрийн үнэлгээгээ 30 хүртэл оноогоор өгөх ба үнэлгээ тавьсан хуудсыг оюутанд буцааж өгөлгүй төгсөлтийн нарийн бичгийн даргад хураалгана уу.*

## ТӨГСӨЛТИЙН АЖЛЫН ЯВЦ

№	Хийж гүйцэтгэсэн ажил	Биелсэн хугацаа	Удирдагчийн гарын үсэг
1	Бүлэг №1. Proxy Re-Encryption схемийн онолын хэсэг	2023-4-28	
2	Бүлэг №2. Серверт шифрлэгдсэн файл хуваалцах судалгаа	2023-4-21	
3	Бүлэг №3. Proxy re-encryption систем хөгжүүлэх	2023-5-18	
4	Бүлэг №4. Ерөнхий дүгнэлт	2023-5-25	

### Ажлын товч дүгнэлт

.....

.....

.....

.....

.....

.....

.....

Удирдагч: ..... /доктор (Ph.D) В.Нямсүрэн/

### ЗӨВШӨӨРӨЛ

Оюутан А.Мягмарцэрэн–н бичсэн төгсөлтийн ажлыг УШК-д хамгаалуулахаар тодорхойлов.

Салбарын эрхлэгч: ..... /доктор (Ph.D) Б.Мөнхбаяр/

ШИНЖЛЭХ УХААН, ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ  
Мэдээлэл, Холбооны Технологийн Сургууль

ШҮҮМЖИЙН ХУУДАС

Мэдээллийн сүлжээ, аюулгүй байдлын салбар–н салбарын төгсөх курсийн оюутан А.Мягмарцэрэн-н "Прокси дахин шифрлэх схемийн туршилтын системийг хөгжүүлэх нь" сэдэвт төгсөлтийн ажлын шүүмж.

1. Төслөөр дэвшүүлсэн асуудал, үүнтэй холбоотой онолын материал уншиж судалсан байдал. Энэ талаар хүмүүсийн хийсэн судалгаа, түүний үр дүнг уншиж тусгасан эсэх.

.....

.....

.....

.....

.....

.....

.....

2. Төслийн ерөнхий агуулга. Шийдсэн зүйлүүд, хүрсэн үр дүн. Өөрийн санааг гарган, харьцуулалт хийн, дүгнэж байгаа чадвар.

.....

.....

.....

.....

.....

.....

.....

3. Эмх цэгцтэй, стандарт хангасан өөрөөр хэлбэл диплом бичих шаардлагуудыг биелүүлсэн эсэх. Төсөлд анзаарагдсан алдаанууд, зөв бичгийн болон өгүүлбэр зүйн гэх мэт /Хуудас дугаарлагдаагүй, зураг хүснэгтийн дугаар болон тайлбар байхгүй, шрифт хольсон, хувилсан зүйл ихээр оруулсан/.

.....

.....

.....

.....

.....

4. Төслөөр орхигдуулсан болон дутуу болсон зүйлүүд. Цаашид анхаарах хэрэгтэй зүйлүүд.

.....

.....

.....

.....

.....

.....

5. Төслийн талаар онцолж тэмдэглэх зүйлүүд.

.....

.....

.....

.....

.....

.....

6. Ерөнхий оноо. (30 оноо)

.....

Шүүмж бичсэн: ..... /магистр Г.Баяр/

Ажлын газар: .....

Хаяг (Утас) .....



## Зорилго

Прокси дахин шифрлэлэх схем судалж, аюулгүй файл хуваалцах систем туршилтын загвар хөгжүүлэх.

## Зорилт

Дээрх зорилгыг хэрэгжүүлэхийн тулд дараах зорилтуудыг дэвшүүлж байна.

- Шифрлэлтийн схемүүдийг судлах
- Файл шифрлэх аргуудыг судлах
- Хөгжүүлэхэд шаардлагатай хэрэглэгдэхүүнийг судлах
- Аюулгүй файл хуваалцах үйлчилгээний загвар гарах
- Хөгжүүлсэн системийг туршиж, ажиллуулах

ШИНЖЛЭХ УХААН, ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ  
Мэдээлэл, Холбооны Технологийн Сургууль

## Хураангуй

Прокси дахин шифрлэх схемийн туршилтын системийг  
хөгжүүлэх нь

А.Мягмарцэрэн  
b190970106@must.edu.com

*Түлхүүр үгс: мэдээллийн аюулгүй байдал, прокси дахин шифрлэлт*

Прокси дахин шифрлэлэх схемийг ашиглан файл хуваалцах үйлчилгээ

# Талархал

Энэхүү дипломын ажлыг бичихэд туслалцаа үзүүлсэн удирдагч багш Н.Чулуунбаатар болон ШУТИС-ийн Мэдээлэл холбоо технологийн сургуулийн Электроникийн салбарын багш нарт талархсанаа илэрхийлье.

# Товчилсон үгс

<b>PRE</b>	<b>P</b> roxy <b>R</b> e- <b>E</b> ncryption
<b>BBS</b>	<b>B</b> laze <b>B</b> leumer <b>S</b> trauss
<b>API</b>	<b>A</b> pplication <b>P</b> rogramming <b>I</b> nterface

# Гарчиг

# Зургийн жагсаалт

# Хүснэгтийн жагсаалт

---

## БҮЛЭГ 1

---

# Өгөгдөл хуваалцах үйлчилгээний тухай



## 1.1 Өгөгдөл хуваалцах үйлчилгээ

Өгөгдөл хуваалцах гэдэг нь ижил өгөгдлийн нөөцийг олон программ, хэрэглэгч эсвэл байгууллагад ашиглах боломжтой болгохыг хэлнэ. Үүнд технологи, практик, хууль эрхзүйн орчин, соёлын элементүүд багтах ба өгөгдлийн бүрэн бүтэн байдлыг алдагдуулахгүйгээр аж ахуйн нэгжүүд хялбар, аюулгүй өгөгдөлд хандах боломжийг олгодог.

Байгууллагын үр ашгийг дээшлүүлж, борлуулагчид болон түншүүдтэй хамтын ажиллагааг дэмжинэ. Хуваалцсан өгөгдлийн эрсдэл, боломжуудын талаар мэдлэгтэй байх нь үйл явцын салшгүй хэсэг юм. [1].

### 1.1.1 Өгөгдөл хуваалцах технологиуд

Өгөгдөл хуваалцах олон технологи байдагаас зарим технологиудаас дурдвал.

- **Өгөгдлийн агуулах (Data warehousing)** нь нэг буюу хэд хэдэн ялгаатай эх сурвалжийг нэгтгэсэн төвлөрсөн агуулах юм. Архитектур нь шатлалаас бүрддэг. Дээд давхарга нь тайлагнах, дүн шинжилгээ хийх, үр дүнг харуулдаг front-end клиент юм. Дунд шат нь өгөгдөлд хандах, дүн шинжилгээ хийхэд ашигладаг аналитик механизмаас бүрдэнэ. Доод шат нь өгөгдлийг ачаалах, хадгалах өгөгдлийн сангийн сервер юм. Дээд болон дунд түвшний програмууд нь доод давхаргад хадгалагдсан нийтлэг өгөгдлийн багцыг хуваалцах боломжтой.
  - Олборлох, хувиргах, ачаалах суурилсан (ETL based data warehouse)
  - Олборлох, ачаалах, хувиргах суурилсан (ELT based data warehouse)
- **Хэрэглээний программчлалын интерфэйс (API)** нь программ хангамжийн бүрэлдэхүүн хэсгүүд тодорхой протоколуудыг ашиглан хоорондоо харилцах боломжийг олгодог механизм юм. Интерфэйс нь хоёр программын хоорондох үйлчилгээний тохиролцоо гэж үзэж болно. Энэхүү тохиролцоо нь хэрхэн харилцах хүсэлт болон хариултыг тодорхойлдог. Хандалтыг нарийн тодорхойлж болдог ба хэрэглэгчид яг ямар өгөгдөл хүсэж болохыг зааж өгдөг.
  - SOAP APIs XML ашигладаг ба уян хатан бус.
  - RPC APIs нь алсаас өөр сервер болон сервисийн функцийг дуудаж ажиллуулдаг.
  - Websocket APIs нь холболт үүсгэж сервер болон хэрлэгч аль ч чиглэлд нэг холболтоор дамжуулах боломжтой.
  - REST APIs нь уян хатах PUT DELETE гэх мэт хүсэлт илгэдэг.
- **Холбооны сургалт (Federated learning)** нь тархсан өгөгдлийг багц дээр хиймэл оюун ухааныг сургах боломжийг олгодог. Бүх өгөгдлийг нэг дор цуглуулж нэгтгэхийн оронд тус тусдаа төхөөрөмж дээр хадгалж зөвхөн загварын шинэчлэлтүүдийг төв сервер рүү илгээдэг.
- **Блокчейн технологи** нь сүлжээн дотор ил тод мэдээлэл солилцох боломжийг олгодог өгөгдлийн сангийн дэвшилтэт механизм юм. Өгөгдлийг гинжин хэлхээнд холбосон блокуудад хадгалдаг. Сүлжээнээс зөвшилцөлгүйгээр гинжийг устгах эсвэл өөрчлөх боломжгүй.

- **Өгөгдөл солилцох платформууд**

Нээлттэй өгөгдлийн платформууд нь өөр өөр өгөгдлийн багцыг нийтийн хэрэгцээнд ашиглах боломжийг олгодог. Ихэвчлэн өгөгдлийн менежмент, өгөгдлийн аюулгүй байдал, өгөгдөл нэгтгэх, өгөгдөл хуваалцах, хамтран ажиллах зэрэг олон төрлийн функцуудыг санал болгодог.

### 1.1.2 Файл хуваалцах

Олон хэрэглэгч эсвэл төхөөрөмж нэг файл эсвэл багц файлд хандах боломжийг олгох практикийг хэлнэ. Файл хуваалцахыг уламжлалт болон орчин үеийн янз бүрийн арга технологи ашиглан хийж болно.

#### Уламжлалт

- Физик зөөвөрлөгч: Файлуудыг CD, DVD эсвэл USB гэх мэт физик медиа ашиглан хуваалцаж болно. Энэ арга нь интернэтийн хандалт хязгаарлагдмал эсвэл боломжгүй үед файл хуваалцахад тустай.
- Сүлжээгээр файл хуваалцах: Файлуудыг Server Message Block (SMB) эсвэл Network File System (NFS) зэрэг технологийг ашиглан дотоод сүлжээгээр хуваалцаж болно. Энэ арга нь байгууллага дотор эсвэл гэрийн сүлжээн дэх төхөөрөмжүүдийн хооронд файл хуваалцахад хэрэгтэй.

#### Орчин үеийн

- Клоуд сан: Dropbox, Google Drive эсвэл OneDrive зэрэг үүлэн хадгалах үйлчилгээ нь хэрэглэгчдэд үүлэн доторх файлуудыг хадгалах, бусадтай хуваалцах боломжийг олгодог. Энэ арга нь өөр өөр төхөөрөмжид байршил хамаарахгүй файл хуваалцахад тустай бөгөөд интернэт холболттой хаанаас ч хандах боломжтой.
- Файл дамжуулах үйлчилгээ: WeTransfer, Hightail эсвэл Filemail зэрэг файл дамжуулах үйлчилгээ нь хэрэглэгчдэд том хэмжээний файлуудыг бусдад хурдан бөгөөд хялбар илгээх боломжийг олгодог. Энэ арга нь байгууллагаас гадуурх хүмүүстэй файл хуваалцах эсвэл имэйл хавсралтын хязгаарт хүрсэн үед хэрэгтэй.
- Per-to-peer файл хуваалцах: Peer-to-peer (P2P) файл хуваалцах нь хэрэглэгчдэд төвлөрсөн сервер ашиглахгүйгээр шууд бие биетэйгээ файл хуваалцах боломжийг олгодог. P2P файл хуваалцах нь ихэвчлэн кино, программ хангамж гэх мэт том файлуудыг хуваалцахад ашиглагддаг боловч бусад төрлийн файлуудыг хуваалцахад ашиглаж болно.

Файл болон өгөгдөл хуваалцах нь олон давуу талтай ч өгөгдлийг эрсдэлд оруулдаг.

## 1.2 Өгөгдлийн аюулгүй байдал

Өгөгдлийн аюулгүй байдал гэдэг нь дижитал мэдээллийг зөвшөөрөлгүй хандах, өөрчлөх, хулгайлахаас хамгаалах үйл ажиллагаа юм. Физик төхөөрөмжийн хамгаалалтаас эхлээд хандалтын удирдлага, программ хангамжийн логик аюулгүй байдал мэдээллийн аюулгүй байдлын бүх талыг хамарсан ойлголт юм. [2]

Нууц мэдээлэл, санхүүгийн чадамж бичиг баримт зэргийг буруу зорилгоор ашиглах боломжтой. Байгууллагын хувьд хэрэглэгчдийнхээ мэдээллийг алдаж буруу гарт орохоос сэргийлж хамгаалах ёстой. Халдлагад өртөх, мэдээлэлээ алдах

нь тухайн байгууллагын нэр хүнд нь халтай ба хэрэглэгчдийн итгэлийг алдах аюултай.

### Өгөгдөл хуваалцах эрсдэлүүд

- **Нууцлалыг задруулах**

Хувийн нууцыг алдагдуулахгүйгээр өгөгдлийг хуваалцахын тулд шифрлэлт, засварлах зэрэг нууцлалыг хамгаалах технологи нь өгөгдлийг аюулгүй хуваалцах боломжийг олгодог.

- **Өгөгдлийн буруу тайлбар**

Өгөгдөл бэлтгэгч болон хэрэглэгчдийн хоорондын харилцаа холбоо дутмагаас буруу тайлбар гарч болзошгүй. Шинжээчид тайлан, үр дүнг тайлбарлахдаа буруу таамаглал дэвшүүлж болно. Жишээлбэл, тухайн сард үйлчлүүлэгчдийн захиалга багассан нь маркетингийн төсөв багатай холбоотой байж болох ч бодит шалтгаан нь бүтээгдэхүүний бэлэн байдлын саатал байж болох юм.

- **Өгөгдлийн чанар муудах**

Давхардсан эсвэл дутмаг чанар муутай өгөгдөл авах эрсдэлтэй.

### Өгөгдлийг аюулгүй хуваалцах төрлүүд

- **Шифрлэлт** нь түлхүүр нууц үггүйгээр өгөгдлийг унших боломжгүй бологдог ба криптографын алгоритмуудыг ашиглан энгийн текстийг шифрлэх үйл явц юм. Энэ нь халдагчид өгөгдөлд нэвтэрсэн байсан ч зохих итгэмжлэлгүйгээр үүнийг уншиж чадахгүй гэдгийг баталгаажуулахад тусалдаг.
- **Хандалтын удирдлага** нь нууц өгөгдөлд хэн хандах эрхтэй болохыг тэдний үүрэг, зөвшөөрлийн түвшинд үндэслэн хязгаарладаг. Үүнд нууц үг, биометрийн баталгаажуулалт, хамгаалалтын токен зэрэг арга хэмжээ багтана.
- **Нөөцлөх, сэргээх** үйл явц нь аюулгүй байдлын зөрчил эсвэл өгөгдөл алдагдсан тохиолдолд сэргээх боломжтой байхын тулд мэдээллийн хуулбарыг үүсгэх, хадгалах явдал юм.
- **Физик аюулгүй байдал** нь өгөгдөл хадгалах төхөөрөмж болон физик хандалтыг хамгаалахын тулд түгжээтэй хаалга, хамгаалалтын камер зэрэг физик хамгаалалтын арга хэмжээг ашигладаг.
- **Өгөгдөл устгах** Өгөгдлийг устгах нь хамгийн аюулгүй хэдий дахин ашиглах боломжгүй. Ихэвчлэн дахин ашиглахгүй өгөгдлийн дарж бичих зэргээр устгадаг.
- **Өгөгдлийн далдлах** нь нууц мэдээллийг анхны өгөгдлийн бүтцийг хадгалан зөвшөөрөлгүй хэрэглэгчдэд ашиглах боломжгүй болгож буй хуурамч мэдээллээр солих явдал юм.

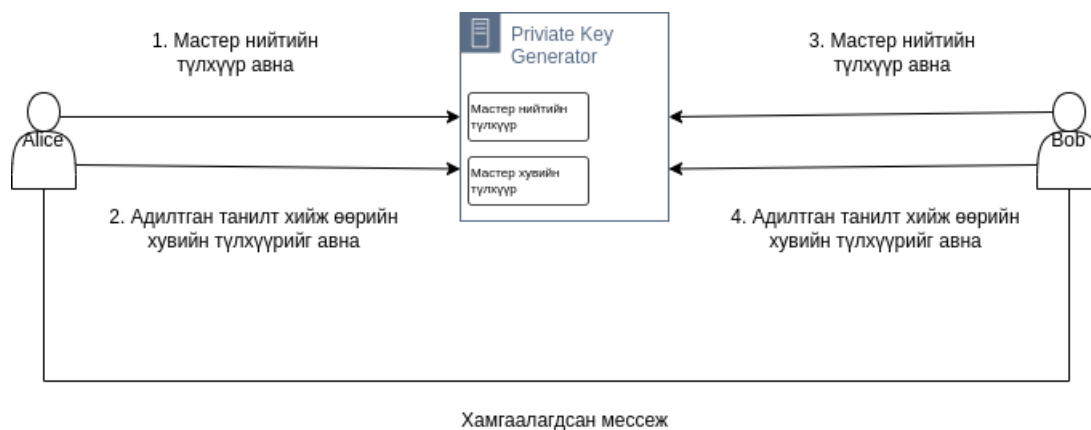
## 1.3 Шифрлэх схемүүд

### Танилтад суурилсан шифрлэлт (IBE)

Тэмдэгт мөр зэрэг мэдэгдэж буй утгаас нийтийн түлхүүр үүсгэх боломжийг олгодог. Итгэмжлэгдсэн гуравдагч тал түлхүүрүүдийг үүсгэж өгдөг(PKG). Хувийн түлхүүр үүсгэгч (PKG) итгэмжлэгдсэн гуравдагч тал холбогдох хувийн түлхүүрүүдийг

үүсгэдэг. PKG эхлээд мастер нийтийн түлхүүрийг нийлтэй тавьж, мастер хувийн түлхүүрийг хадгална. Аль ч тал мастер нийтийн түлхүүр, таних утгыг ашиглан тохирох нийтийн түлхүүрийг гаргаж авах боломжтой. Харгалзах хувийн түлхүүрийг авахын тулд мастер түлхүүрээр гаргаж авсан таних түлхүүрийг ашиглана. [3]

1. Бэлтгэл үе: PKG нь өөрийн мастер түлхүүрүүдийг үүсгэнэ.
2. Алис нийтийн мастер түлхүүрийг авна. Өөрийн хувийн түлхүүрыг авна.
3. Боб-ийн имэйл гэх мэт өвөрмөц мэдээллээр Боб-ын нийтийн түлхүүрийг авч шифрлэлт хийн явуулна.
4. Боб PKG-ээс өөрийн хувийн түлхүүрийг авч шифрийг тайлж авна.



ЗУРАГ 1.1: Танилтад суурилсан шифрлэлт

IBE-дээр суурилсан алгоритмууд

- Boneh–Franklin (BF-IBE)
- Sakai–Kasahara (SK-IBE)
- Boneh–Boyen (BB-IBE)

Давуу талууд

- Тодорхойгүй олон хэрлэгч үед давуу талтай
- Мөн нийтийн түлхүүрийг хэрэглээг халж орлох боломжтой
- Мөн цаг минут зэрэг ашиглан хандах эрэхийг нарийн тодорхойлж болно.

## Шинж чанарт суурилсан шифрлэлт (ABE)

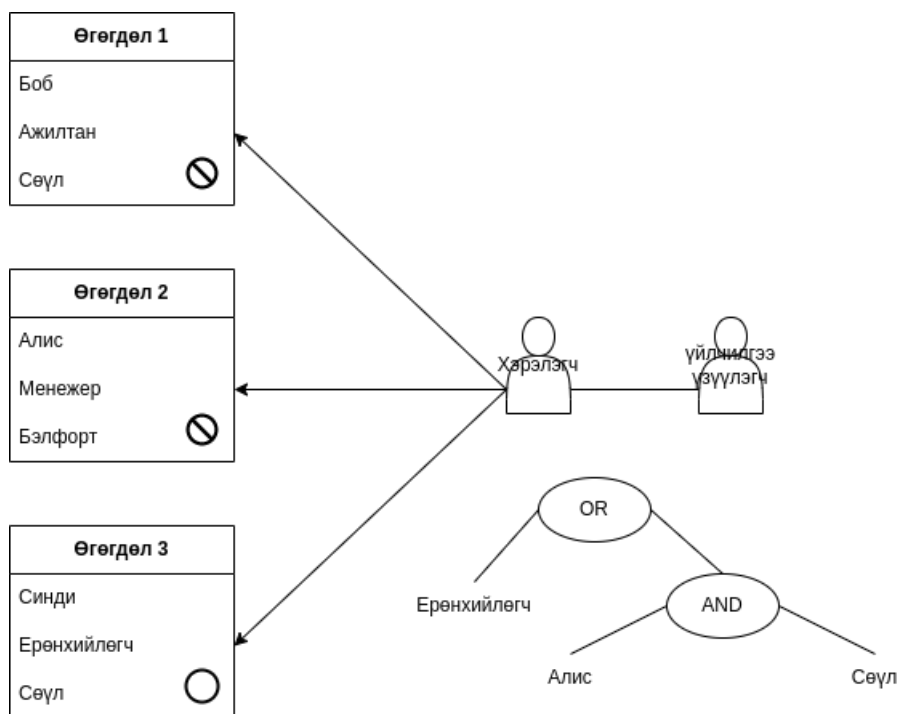
IBE-тэй ерөнхийдөө төстэй. Шинж чанаруудаар бүлэглэж зөвхөн нэг хэрлэгчийн түлхүүр ашиглахгүй олон хүн тайлах боломжтой. Үндсэн хоёр төрөлтэй. Түлхүүр-Дүрэмийн шинж чанарт суурилсан шифрлэлт (KP-ABE) болон Шифртескт-Дүрэмийн шинж чанарт суурилсан шифрлэлт (CP-ABE).

ABE ерөнхий санаа нь сайн болов ч сул талуудтай.

- Түлхүүр зохицуулалт
- Түлхүүр хадгалах
- Түлхүүр хүчингүй болгох

## КР-ABE

Хэрэглэгч бүр шинж чанруудтай. Хэрэглэгч нь хандалтын модыг өөртөө агуулдаг тул бүх хэрэглэгч өгөгдөл рүү хандах эрхтэй байдаг.



ЗУРАГ 1.2: Танилтад суурилсан шифрлэлт

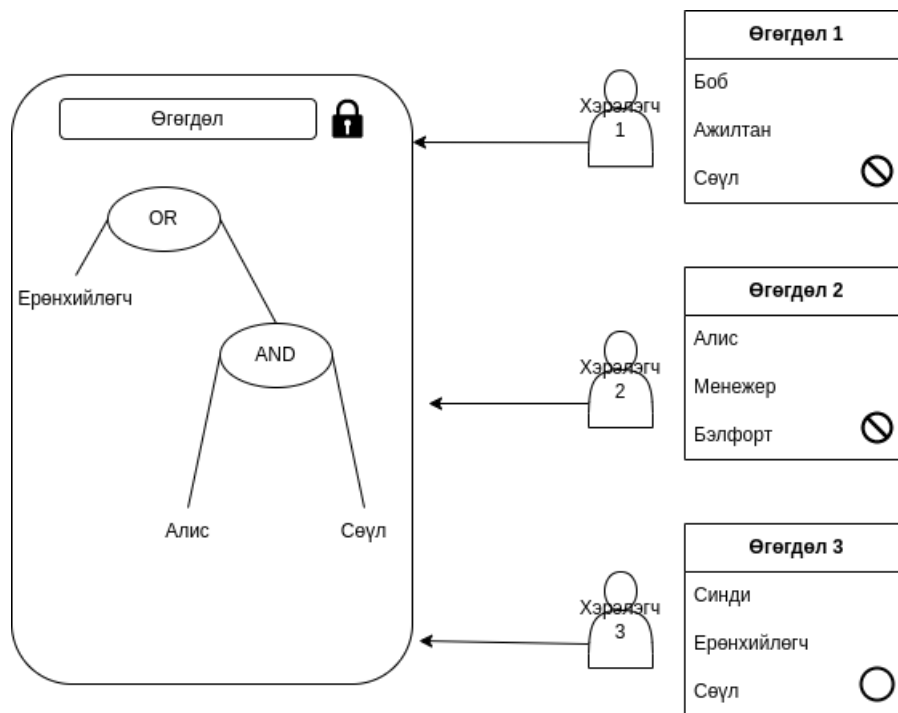
## СР-ABE

КР-ABE-ээс ялгаатай зүйл нь шифр нь өрөө хандалтын модыг агуулдаг. КР-ABE-ээс ялгаатай нь шифрлэгдсэн текст нь хандалтын модыг агуулдаг. Хэрэв хэрэглэгчийн шинж чанар нь шифрлэгдсэн текст дэх хандалтын модны нөхцөлийг хангаж байвал шифрлэгдсэн өгөгдлийг тайлах боломжтой. Хандалтын мод нь шифрлэгдсэн өгөгдөлд байгаа тул өгөгдлийн хандалтыг хянах боломжтой.

## Гомоморф шифрлэлт (HE)

Энэ нь шифрлэгдсэн өгөгдлийг тайлахгүйгээр тооцоолол хийх боломжийг олгодог шифрлэлтийн төрөл юм. Аюулгүй үүлэн тооцоолол, цахим санал хураалт, эмнэлгийн бүртгэл гэх мэт төрөл бүрийн хэрэглээнд хэрэглэж болдог. [4]

- Хэсэгчилсэн гомоморф шифрлэлт (PHE). PHE-д "хэсэгчилсэн" гэдэг нь зөвхөн сонгосон математик функцийг шифрлэгдсэн утгууд дээр боловсруулах боломжтой. Зөвхөн нэг үйлдлийг нэмэх эсвэл үржүүлэхийг шифрлэгдсэн текст дээр хязгааргүй олон удаа хийхийг зөвшөөрдөг.
- Зарим ижил төстэй шифрлэлт (SHE). "Somewhat" нь PHE-ээс илүү ерөнхий шинж чанартай бөгөөд энэ нь нэмэх, үржүүлэх бүхий гомоморф үйлдлийг дэмждэг. Гэхдээ энд байгаа гол сул тал нь зөвхөн цөөн тооны үйлдлийг гүйцэтгэх боломжтой юм.



ЗУРАГ 1.3: Танилтад суурилсан шифрлэлт

- Бүрэн гомоморфик шифрлэлт (FHE). PHE болон SHE нь хязгаарлагдмал ажиллагаатай бол бүрэн гомоморф шифрлэлт нь шифрлэгдсэн текстэн дээр хязгаарлалтгүйгээр нэмэх болон үржүүлэх үйлдлийг хоёуланг нь ашиглах чадвартай.

Хямд үүлэн тооцоолол, үүлэн хадгалах систем нь аж ахуйн нэгжүүд болон хувь хүмүүсийн өгөгдлөө ашиглах, удирдах арга хэлбэрийг үндсээр нь өөрчилсөн. AES гэх мэт уламжлалт шифрлэлтийн аргууд нь маш хурдан бөгөөд өгөгдлийг шифрлэгдсэн хэлбэрээр хялбархан хадгалах боломжийг олгодог. Гэсэн хэдий ч, шифрлэгдсэн өгөгдөл дээр энгийн дүн шинжилгээ хийхийн тулд үүлэн сервер нууц түлхүүрт хандах шаардлагатай бөгөөд энэ нь аюулгүй байдлын асуудалд хүргэдэг, эсвэл өгөгдөл эзэмшигч нь өгөгдлийг татаж авах, тайлах, локал байдлаар ажиллуулах шаардлагатай. зардал ихтэй, логистикийн сорилтыг бий болгоно. Клоуд нь шифрлэгдсэн өгөгдөл дээр шууд ажиллаж, зөвхөн шифрлэгдсэн үр дүнг өгөгдлийн эзэнд буцааж өгөх боломжтой тул гомоморф шифрлэлтийг энэ хувилбарыг ихээхэн хялбарчлахад ашиглаж болно. Хэрэглээний илүү төвөгтэй хувилбарууд нь гуравдагч этгээдийн ажиллах боломжтой хувийн өгөгдөл бүхий олон талуудыг оролцуулж, үр дүнг нэг буюу хэд хэдэн оролцогчид шифрлэхээр буцааж өгч болно.

#### Давуу талууд

- Найдваргүй гуравдагч орчинд өгөгдөл найдвартай, нууцлалтай хэвээр байна. Өгөгдөл нь үргэлж шифрлэгдсэн хэвээр байх бөгөөд энэ нь нууц мэдээлэл алдагдуулах магадлалыг бууруулдаг.
- Бүрэн гомоморф шифрлэлтийн схемүүд нь квант халдлагын эсрэг тэсвэртэй.

#### Сул талууд

- Тооцооллын хурд багатай
- Нарийвчлал багатай

## 1.4 Файл шифрлэх хадгалах

Файлыг тэгш хэмт болон тэгч бус шифрлэлтээр аль алингаар нь шифрлдэг. Тэгш хэмт шифрлэлт нь тэгш бус хэмт шифрлэлтээс хувьд хурдан боловч түлхүүр дамжуулах зэрэгт асуудал үүсдэг. Ихэвчлэн тэгш бус шифрлэлтээр файлыг шифрлдэг ба тэгш хэмт шифрлэлтээс илүү хурдан байдаг.

### Нийтлэг ашигладаг алгоритмууд

- Нарийвчилсан шифрлэлтийн стандарт (AES): AES нь тэгш хэмт шифрлэлтийн алгоритм бөгөөд хамгийн найдвартай гэж тооцогддог.
- Гурвалсан өгөгдөл шифрлэлтийн стандарт (3DES): 3DES нь аюулгүй гэж тооцогддог хуучин тэгш хэмтэй шифрлэлтийн алгоритм юм. Энэ нь ихэвчлэн өндөр хамгаалалттай байх шаардлагагүй өгөгдлийг шифрлэхэд ашиглагддаг.
- Ривест-Шамир-Адлеман (RSA): RSA нь бусадтай хуваалцах шаардлагатай өгөгдлийг шифрлэхэд ихэвчлэн ашиглагддаг тэгш бус шифрлэлтийн алгоритм юм. Үүнийг мөн тоон гарын үсэг үүсгэхэд ашигладаг бөгөөд үүнийг баримт бичиг эсвэл мессежийн жинхэнэ эсэхийг шалгахад ашиглаж болно.

### Файлын системийг шифрлэх

Файл системийн түвшинд шифрлэлтийг файлд суурилсан шифрлэлт (FBE) эсвэл файл/хавтас шифрлэлт гэдэг нэрлдэг дискний шифрлэлтийн нэг төрөл юм.[5]

#### Файл системийн шифрлэлтийн төрөл

- Криптограф файлын систем
- Ерөнхий зориулалт бүхий файлын шифрлэлт системүүд

#### Давуу талууд

- Файлд суурилсан уян хатан түлхүүрийн удирдлага гэдэг нь файл бүрийг тусдаа түлхүүрээр шифрлэх боломжтой.
- Шифрлэгдсэн файлын тус тусд нь удирдах нь шифрлэгдсэн файлыг бүхэлд нь засаж өөрчлөхийн оронд зөвхөн өөрчлөгдсөн хэсгийг л өөрчлөх боломжтой.
- Хандалтын нийтийн түлрүүр ашиглах хянах боломжтой.

### Криптограф файлын систем

Криптограф файлын системүүд нь шифрлэлт, аюулгүй байдлын үүднээс тусгайлан бүтээгдсэн (ерөнхий зориулалтын бус) файлын системүүд юм. Тэд ихэвчлэн мета өгөгдлийг оруулаад агуулсан бүх өгөгдлийг шифрлэдэг. Диск дээрх формат болон өөрийн блокийн хуваарилалт хийдэггүй одоо байгаа файлын системүүдийн дээр байрладаг.

### Ерөнхий зориулалт бүхий файлын шифрлэлт системүүд

Криптограф файлын систем болон бүрэн дискний шифрлэлтээс ялгаатай нь файлын системийн түвшинд шифрлэлт хийдэг ерөнхий зориулалтын файлын системүүд юм. Файлын нэр, хэмжээ, өөрчлөлтийн цагийн тэмдэг гэх мэт файлын системийн мета өгөгдлийг ихэвчлэн шифрлэдэггүй.

## 1.5 Бүлгийн Дүгнэлт

Өгөгдөл хуваалцах болон өгөгдөлийн аюулгүй байдлын судалсан. Орчин үеийн шифрлэлтийн схемүүдийн судлаж файл шифрлэх аргуудыг судалсан.



---

## БҮЛЭГ 2

---

Прокси дахин шифрлэлтэд суурилсан  
файл хуваалцах систем

## 2.1 Прокси дахин шифрлэлт

Прокси дахин шифрлэлт нь нийтийн түлхүүрээр шифрлэсэн өгөгдлийг дахин шифрлэж өөр хувийн түлхүүрээр тайлах боломжийг олгодог.

Мамбо болон Окамото шифрийг тайлан дараа нь шифрлэх уламжлалт аргыг сайжруулах зорилгоор анх гаргаж ирсэн.

1998 онд Blaze, Bleumer, Strauss (BBS) нар "atomic proxy cryptography" гэсэн ойлголтыг санал болгосон бөгөөд үүнд хагас итгэмжлэгдсэн прокси нь үндсэн энгийн текстийг харалгүйгээр Алисын шифрийг Бобын шифр текст болгон хувиргадаг. El Gamel дээр суурилсан схем ба прокси буюу гуравдагч талийн тусламжтайгаар шифрийг дамжуулах зорилготой. [6]

Прокси дахин шифрлэлт үйл явц нь гурван хэсгээс тогтоно.

- **Төлөөлөгч:** Прокси дахин шифрлэлт ашиглан шифрийг тайлах эрхээ өөр хүнд шилжүүлэх өгөгдлийн эзэн. Дахин шифрлэлтийн түлхүүр үүсгэж, прокси руу илгээдэг. Төлөөлөгчийг ихэвчлэн "Алис" гэж нэрлэдэг.
- **Орлогч:** Төлөөлөгчийн шифрийг тайлах эрхтэй хүн. Ихэвчлэн "Боб" гэдэг нэртэй байдаг.
- **Прокси:** Дахин шифрлэх шифрийг дамжуулах гуравдагч тал.

**Прокси дахин шифрлэлт үндсэн хоёр төрөлтэй.**

- Нэг чиглэлт (Unidirectional PRE)
- Хоёр чиглэлт (Bidirectional PRE)

Мөн олон дахин шифрлэх боломжтой болон нэг удаагийн шифрлэх боломжтой гэж хуваадаг.

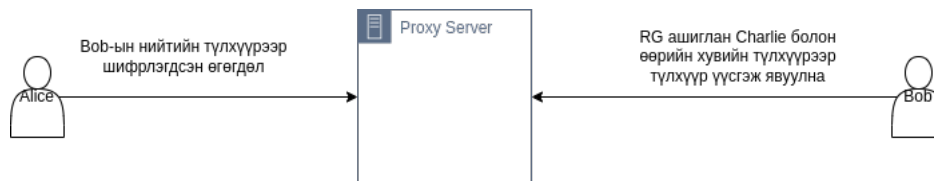
Прокси дахин шифрлэх схем дараах хэсгүүдээс тогтоно.

- **Нийтийн параметрууд:** Нийтлэг нийтэд байдаг параметрууд. Түлхүүрийн урт зэрэг.
- **Нийтийн болон нууц түлхүүрийн хослол:** Хэрэглэгч бүр нийтийн/нууц түлхүүрийг үүсгэдэг. Нийтийн түлхүүрийг нийтэд тавьж, харин хувийн түлхүүр нь зөвхөн хэрэглэгч өөртөө үлдээнэ. Боб Алис руу мессеж илгээхдээ Алисын нийтийн түлхүүрийг ашиглан шифрлэж, Алис түүний хувийн түлхүүрийг ашиглан тайлна.
- **Төлөөлөгчийн түлхүүрүүд:** Хэрэглэгч бүр энд  $rk_{Alice} \rightarrow Bob$  гэж тэмдэглэсэн дурын төлөөлөгчид (дахин шифрлэх) түлхүүр үүсгэж болно.  $rk_{Alice} \rightarrow Bob$  тулд Алис прокси дахин шифрлэлтийн алгоритмыг ашиглан хувийн түлхүүрээ Бобын нийтийн түлхүүртэй хослуулан түлхүүр гаргаж авна. Гаргаж авсан түлхүүрээр дахин шифрлэж төлөөлөгчийн хувийн түлхүүрээр тайлах боломжтой болно.
- **Шифрлэгдсэн текст:** Хэрэглэгч ямар нэгэн нийтийн түлхүүрээр мессежийг (энгийн текст) шифрлэх үед шифр текст үүснэ.

Нэг чиглэлт  $PRE (KE, RG, E, R, D)$  хэсгүүдээс тогтоно.

1. Алис, Боб болон Чарли хувийн болон нийтийн түлхүүрийг үүсгэнэ.  $(KE)$
2. Алис Боб-д зориулж өгөгдлөө шифрлэж серверт байршуулна.

3. Боб Алис-ын өгөгдлийг Чарли-тай хуваацлахын тулд  $RE(pkB, skB, pkC, skC)$  шифрлэж серверт байршуулна. Чарлигийн хувийн заавал шаардахгүй үүсгэж болно.
4. Боб RE-г ашиглаж үүсгэсэн түлхүүрийг серверт явуулж Алисын файлыг дахин шифрлэж Чарли тайлах боломжтой болно.



ЗУРАГ 2.1: Прокси дахин шифрлэх

Давуу талууд:

- Нууцлалыг сайжруулна: PRE нь оролцогч талуудын хувийн мэдээллийг задруулахгүйгээр өгөгдлийг хуваалцахыг зөвшөөрснөөр нууцлалыг сайжруулдаг. Талууд нууцаар эсвэл хувийн нууц мэдээллийг задруулахгүйгээр мэдээллээ хуваалцахыг хүссэн тохиолдолд ашиглах боломжтой.
- Илүү уян хатан шифрлэсэн өгөгдлийг өргөн хүрээтэй тараах боломжтой

Сул талууд:

- Проксид итгэх: PRE нь дахин шифрлэлтийг гүйцэтгэхэд гуравдагч талын прокси дээр тулгуурладаг ба схемийн аюулгүй байдал нь прокси талаас хамаарна.
- Хязгаарлагдмал өргөтгөх чадвар: PRE нь өргөтгөх чадварын хувьд хязгаарлагдмал байж болно. Учир нь хэрэглэгчдийн тоо нэмэгдэхийн хэрээр олон талыг дэмжихэд шаардлагатай дахин шифрлэлтийн түлхүүрүүдийн тоо хурдацтай өсөх болно. Энэ нь гол менежментийг төвөгтэй болгож, удирдахад хэцүү болгодог.
- Potential for replay attacks: PRE нь халдагч хариуг зогсоож хандах эрхийг өөрт ашигтай солих боломжтой.
- Хүчингүй болгоход хүндрэлтэй: PRE дахь өгөгдөлд хандах эрхийг цуцлах нь ялангуяа олон тал оролцсон тохиолдолд төвөгтэй. Хэрэв аль нэг талын дахин шифрлэлтийн түлхүүр алдагдсан бол бусад талуудын мэдээлэлд хандах эрхэд нөлөөлөхгүйгээр өгөгдөлд хандах эрхийг цуцлах нь хэцүү юм.
- Хязгаарлагдмал хэрэглээ: PRE нь харьцангуй шинэ бөгөөд шинээр гарч ирж буй технологи хэвээр байгаа бөгөөд илүү уламжлалт шифрлэлтийн схемүүдтэй харьцуулахад хэрэглээ нь хязгаарлагдмал байдаг. Энэ нь технологийг хэрэгжүүлэх, удирдах туршлагатай мэргэжилтнүүд бага байдаг.

### Прокси дахин шифрлэх схемийн ерөнхий хэрэглээ

- **Аюулгүй өгөгдөл хуваалцах:** Өгөгдлийн олон хэрлэгчид найдвартай хуваалцахад ашигладаг. Өгөгдлийн эзэмшигч нь өгөгдлийг өөрийн түлхүүрээр

шифрлэж, дахин шифрлэж прокси руу шилжүүлдэг. Дараа нь прокси нь хүлээн авагчийн түлхүүрээр шифрлэгдсэн өгөгдлийг хувиргаж, өгөгдөл эзэмшигчийн оролцоогүйгээр шифрийг тайлж, өгөгдөлд хандах боломжийг олгодог.

- **Cloud Storage:** Үүл хадгалах системд хадгалагдсан мэдээллийн аюулгүй байдлыг сайжруулахын тулд прокси дахин шифрлэлтийг ашиглаж болно. Нууц мэдээллийг үүлэн үйлчилгээ үзүүлэгч рүү шууд оруулахын оронд өгөгдлийг эзэмшигчийн түлхүүрээр шифрлэж, прокси түлхүүрээр дахин шифрлэх боломжтой. Дараа нь прокси нь дахин шифрлэгдсэн өгөгдлийг үүлэн дотор хадгалдаг. Энэ нь үүлэн үйлчилгээ үзүүлэгч нь зөвхөн дахин шифрлэгдсэн өгөгдөлд хандах эрхтэй бөгөөд анхны контентыг уншиж чадахгүй байх боломжийг олгодог.
- **Түлхүүр эскроу:** Түлхүүр эскроу гэдэг нь криптограф түлхүүрийг итгэмжлэгдсэн гуравдагч этгээдэд хадгалах практикийг хэлнэ. Энэ тохиолдолд анхны өгөгдөл нь эзэмшигчийн түлхүүрээр шифрлэгдсэн бөгөөд дахин шифрлэлтийн түлхүүрнийг проксид хадгалагдана. Хэрэв эзэмшигч нь түлхүүрээ алдсан эсвэл өөр төхөөрөмжөөс хандах шаардлагатай бол прокси нь өгөгдлийг шинэ түлхүүрээр дахин шифрлэх боломжтой бөгөөд ингэснээр эзэмшигчид дахин хандах эрх олгоно.
- **Аюулгүй Имэйл Холбоо:** Илгээгч нь хувийн түлхүүрээ шуудангийн сервер эсвэл бусад зуучлагчтай хуваалцахын оронд өөрийн түлхүүрээр имэйлээ шифрлэж, дахин шифрлэх ажиллагааг прокси руу шилжүүлэх боломжтой. Дараа нь прокси нь хүлээн авагчийн түлхүүрээр имэйлийг дахин шифрлэх боломжтой бөгөөд зөвхөн хүлээн авагч нь мессежний кодыг тайлж унших боломжтой.

## 2.2 Хөгжүүлэх технологи, хэл сонгох

Прокси дахин шифрлэлт схемийг ашиглан файл хуваалцах систем хөгжүүлэхээр болсон. Хөгжүүлэлтэнд хэрэгтэй хэл болон технологи сонгосон.

Систем хоёр хэсгээс тогтох ба. API сервер болон хэрлэгч талын программ. Пайтон маш олон нэмэлт сангууд байдаг маш том нийгэмлэгтэй (community) хэл. Бичиглэл хялбар олон давуу талтай тул пайтон хэлийг сонгосон. Үүнд:

### Flask framework

Пайтон хэл дээр бичсэн веб хөгжүүлэхэд зориулагдсан фреймворк юм. Хөнгөхөн олон нэмэлт сан шаардахгүй хэрэгтэй сангуудаа суулгаад хөгжүүлэх боломжтой. Сурахад хялбар өөрийн хүссэн загвараар загварчилж хийх боломжтой. RESTful API гаргахад хялбар. Хэрэгтэй гэвэл нэмэлт сан ORM зэрэг өөр сангууд суулгаж хамт ашиглах боломжтой.

```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5 @app.route("/")
6 def hello_world():
7     return "<p>Hello, World!</p>"
```

```

8 @app.get("/")
9     def hello_world():
10         return "<p>Hello, <u>get</u> request!</p>"

```

app.route endpoint бичих боломжтой маш хялбар мөн хурдан бичих боломжтой.

## Tkinter

Хэрэглэгчийн интерфейс (GUI) үүсгэхэд ашигладаг Python номын сан юм. Tcl/Tk GUI дээр суурилсан. Линукс виндовс зэрэг олон үйлдлийн системийг дэмждэг. Нэмэлт сангуудтай ажиллах боломжтой.

```

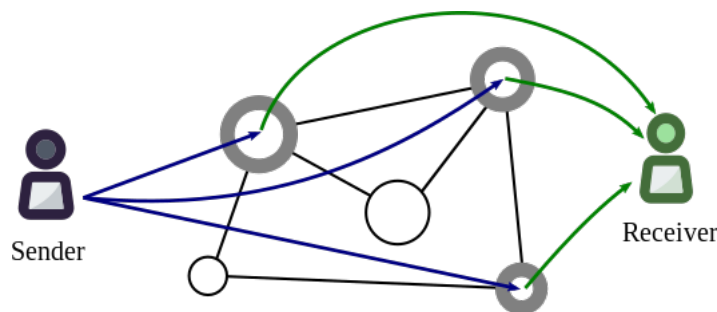
1 import tkinter as tk
2
3 root = tk.Tk()
4 # Create a label widget.
5 label = tk.Label(root, text="Hello, <u>world!</u>")
6 # Create a button widget.
7 button = tk.Button(root, text="Click <u>me!</u>")
8 # Add the label and button widgets to the root window.
9 label.pack()
10 button.pack()
11 # Start the main loop.
12 root.mainloop()

```

Энгийн бичхэд хялбар нийгэмлэг сайн.

## pyUmbral

Прокси дахин шифрлэлтийг файтон дээр хэрэгжүүлсэн пайтоны сан юм. OpenSSL болон Cryptography.io ашигласан.



ЗУРАГ 2.2: pyUmbral

ECIES-KEM болон BBS98 санаа авсан нэг чиглэлтэй дахин шифрлэлтийн сан. pyUmbral функцуудийг ерөнхийд нь гурав ангилсан.

1. Түлхүүр үүсгэх алгоритмууд.
  - KeyGen()
  - ReKeyGen(skA, pkB, N, t)
2. Encapsulation болон Decapsulation
  - Encapsulate(pkA)

- Decapsulate(skA, capsule)

### 3. Re-Encapsulation болон Fragments Decapsulation.

- ReEncapsulation(kF rag, capsule)
- DecapsulateFrag(skB, cF ragiti=1, capsule):

## Google Cloud Platform

Гуравдагч тал болох сервер проксиг (API) deploy хийж байршуулах шаардлагатай0. Google Cloud нь маш олон давуу талтай ба үнэгүй туршиж үзэх 300 долларын эрхтэй тул сонгосон.

## 2.3 Хөгжүүлэлтийн орчин бэлдэх

Пайтон хэл нь орчин бүрдүүлэхэд хялбар ба виртуал орчин үүсгэж хэрэгтэй сангуудыг татаж суулгах боломжтой. Бүх линукс тархцад пайтон хэл нь суулгалцан байдаг тул хэрэгтэй сангуудыг татаж суулгахад л хангалтай.

Пайтон хэл дээр хөгжүүлэлт хийхийн тулд прокси дахин шифрлэлтийн пайтон дээр сан бичих гэж оролдож үзсэн. JHU-MIT Proxy Re-cryptography Library (PRL) нь хэд хэдэн прокси дахин шифрлэлтийн схемийг хэрэгжүүлсэн нээлттэй эхтэй сан юм. Алгоритмуудыг Жонс Хопкинсийн Их Сургууль болон Массачусетсийн Технологийн Их Сургуулийн судлаачид боловсруулсан бөгөөд Жузеппе Атениезе, Кевин Фу, Мэттью Грин, Сюзан Хохенбергер нарын анхны судалгаанд үндэслэсэн. JHU-MIT Прокси дахин шифрлэлтийн сан нь C/C++ хэл дээр бичигдсэн байсан. Пайтон хэлний setuptools болон C хэлний "python.h"санг ашиглан пайтоны сан бичих гэж оролдож үзсэн. JHU-MIT сан нь MIRCL санг ашигладаг.

PRL сан нь хоёр схемийг хэрэгжүүлдэг.

- **PRE1:** алгоритм нь Decisional Bilinear Diffie-Hellman Assumption (DBDH) дагуу аюулгүй.
- **PRE2:** алгоритм нь Decisional Bilinear Inversion Assumption (DBDHI)-ийн дагуу аюулгүй.

PRE1 схемийг пайтон сан руу хөрвүүлэх гэж оролдсон.

### Ерөнхий бүтэц

```
.
├── example
│   └── example.py
├── external
│   ├── miracl
│   └── proxylib
├── LICENSE
├── README.md
├── setup.py
├── src
│   └── pypre.cpp
└── userguide.pdf
```

- **example** хэсэг тестийн код байрлана.
- **external** хэсэгт PRL сан болон MIRCL сан байна.

- Сангуудыг compile хийж **lib\*.a** файл гаргаж авна.
- **setup.py** файл нь тухайн package-ын мэдээлэл байна.
- **pypre.cpp** C++ функцийг нэмэлт сан \*.a-аас дуудаж пайтон руу хөрвүүлэх хэсэг.

Эх код 2.1: setup.py

```

1 import glob
2 import os
3
4 from setuptools import Extension, setup
5
6 os.environ["CC"] = "g++"
7 os.environ["CXX"] = "g++"
8
9 module = Extension(
10     "pypre",
11     language="c++",
12     sources=[f for f in glob.glob('src/*.cpp')] +
13             ['external/proxylib/proxylib_pre1.cpp'],
14     include_dirs=["external/miracl", "external/proxylib"],
15     library_dirs=["external/miracl", "external/proxylib"],
16     libraries=["miracl", "proxylib"],
17     extra_objects=['external/miracl/libmiracl.a', 'external/proxylib/
18                     libproxylib.a']
19 )
20 setup(name='pypre',
21       license='Apache',
22       author="Myagmartseren",
23       author_email='myagmartseren7@gmail.com',
24       description='Proxy_Re-encryption_Python_library',
25       url='https://github.com/myagmartseren/pre_python',
26       ext_modules=[module],
27       include_dirs=['external/proxylib', 'external/miracl'],
28 )

```

**pypre.py** файл дотор үндсэн C++ кодны санг дуудаж пайтонруу хөрвүүлэх хэсэг байна. Хэрэгтэй толгой файлыг include хийнэ. Proxylib.h болон miracle гэх мэт сангууд нь setup.py хавсарсан сангууд юм.

Эх код 2.2: setup.py

```

1 #define PY_SSIZE_T_CLEAN
2 #include <zzn.h>
3 #include <miracl.h>
4
5 #include <Python.h>
6 #include <proxylib.h>
7 #include <proxylib_pre1.h>

```

Class бүрийг C-ын struct-ыг ашиглаж зарлаж өгнө. Class-ын method-уудыг мөн адил зарлаж өгнө.

Эх код 2.3: setup.py

```

1 typedef struct
2 {
3     PyObject_HEAD
4     ProxyCiphertext_PRE1 *proxyCiphertext_PRE1;
5 } PyProxyCiphertext_PRE1;
6
7 static PyObject* PyPRE1_generate_params(PyObject *self, PyObject *args)
8 {
9     if (PyTuple_Size(args) !=1){
10         PyErr_SetString(PyExc_RuntimeError, "Should_have_CurveParams");
11         return Py_None;
12     }
13
14     PyObject *params = PyTuple_GetItem(args, 0);
15
16     if (!PyCurveParams_Check(params)) {
17         PyErr_SetString(PyExc_RuntimeError, "Failed_to_cast_CurveParams");
18         ;
19         return Py_None;
20     }
21
22     PyCurveParams *pyParams = (PyCurveParams *)params;
23
24     int result = PRE1_generate_params(*pyParams->params);
25     if (!result) {
26         PyErr_SetString(PyExc_RuntimeError, "Failed_to_generate_params");
27         return PyBool_FromLong(result);
28     }
29
30     Py_INCREF(Py_None);
31     return PyBool_FromLong(result);
32 }
33
34 static PyObject* PyPRE1_keygen(PyObject *self, PyObject *args) {
35     if (PyTuple_Size(args) !=3){
36         PyErr_SetString(PyExc_RuntimeError, "Should_have_3_parameters");
37         return Py_None;
38     }
39
40     PyObject *tempParams = PyTuple_GetItem(args, 0);
41     if (!PyCurveParams_Check(tempParams)) {
42         PyErr_SetString(PyExc_RuntimeError, "Failed_to_cast_CurveParams");
43         ;
44         return Py_None;
45     }
46
47     PyCurveParams *pyParams = (PyCurveParams *)tempParams;
48     printf("pyParams->params_address:%p\n", pyParams->params);
49
50     PyObject *tempPublicKey = PyTuple_GetItem(args, 1);

```



```

47     if (!PyProxyPK_PRE1_Check(tempPublicKey)) {
48         PyErr_SetString(PyExc_RuntimeError, "Failed to cast PK_PRE1");
49         return Py_None;
50     }
51     PyProxyPK_PRE1 *pyPublicKey = (PyProxyPK_PRE1 *)tempPublicKey;
52     printf("pyPublicKey->publicKey_address: %p\n", pyPublicKey->publicKey
53         );
54
55     PyObject *tempSecretKey = PyTuple_GetItem(args, 2);
56     if (!PyProxySK_PRE1_Check(tempSecretKey)) {
57         PyErr_SetString(PyExc_RuntimeError, "Failed to cast SK_PRE1");
58         return Py_None;
59     }
60     PyProxySK_PRE1 *pySecretKey = (PyProxySK_PRE1 *)tempSecretKey;
61     printf("pySecretKey->secretKey_address: %p\n", pySecretKey->secretKey
62         );
63
64     int result = PRE1_keygen(*pyParams->params, *pyPublicKey->publicKey,
65                             *pySecretKey->secretKey);
66
67     Py_INCREF(Py_None);
68     return PyBool_FromLong(result);
69 }
70
71 static PyMethodDef module_methods[] = {
72     {"PRE1_generate_params", PyPRE1_generate_params, METH_VARARGS, "
73         Generate curve parameters for the PRE1 scheme."},
74     {"PRE1_keygen", PyPRE1_keygen, METH_VARARGS, "Generate curve
75         parameters for the PRE1 scheme."},
76     {NULL, NULL, 0, NULL}
77 };

```

Классуудыг модулд нэмж өгнө. Мөн статик функцуулийг нэмж өгнө.

Эх код 2.4: setup.py

```

1 static struct PyModuleDef module_def = {
2     PyModuleDef_HEAD_INIT,
3     "pypre",
4     0, // m_doc
5     -1, // m_size
6     module_methods, // m_methods
7     /*
8     NULL, // m_reload
9     NULL, // m_traverse
10    NULL, // m_clear
11    NULL // m_free
12    */
13 };
14
15 PyMODINIT_FUNC PyInit_pypre(void);

```

```

16
17 PyObject *PyInit_pypre(void)
18 {
19     extern PyTypeObject PyCurveParamsType;
20     if (PyType_Ready(&PyCurveParamsType) < 0)
21         return NULL;
22
23     PyObject *module = PyModule_Create(&module_def);
24
25     Py_INCREF(&PyCurveParamsType);
26     if (PyModule_AddObject(module, "CurveParams", (PyObject *)&
27         PyCurveParamsType) < 0)
28     {
29         Py_DECREF(&PyCurveParamsType);
30         Py_DECREF(module);
31         return NULL;
32     }
33
34     extern PyTypeObject PyProxyPK_PRE1Type;
35     if (PyType_Ready(&PyProxyPK_PRE1Type) < 0)
36         return NULL;
37
38     Py_INCREF(&PyProxyPK_PRE1Type);
39     if (PyModule_AddObject(module, "PK_PRE1", (PyObject *)&
40         PyProxyPK_PRE1Type) < 0)
41     {
42         Py_DECREF(&PyProxyPK_PRE1Type);
43         Py_DECREF(module);
44         return NULL;
45     }
46
47     extern PyTypeObject PyProxySK_PRE1Type;
48     if (PyType_Ready(&PyProxySK_PRE1Type) < 0)
49         return NULL;
50
51     Py_INCREF(&PyProxySK_PRE1Type);
52     if (PyModule_AddObject(module, "SK_PRE1", (PyObject *)&
53         PyProxySK_PRE1Type) < 0)
54     {
55         Py_DECREF(&PyProxySK_PRE1Type);
56         Py_DECREF(module);
57         return NULL;
58     }
59     // PyModule_AddObject(module, "__version__", PyUnicode_FromString(
60         VERSION));
61
62     if (PyErr_Occurred())
63         PyErr_SetString(PyExc_ImportError, "pypre: _init_ failed");
64
65     return module;

```

62 }

PRE1 хөрвүүлж байсан боловч цаг хугацааны хувьд амжихгүй болсон тул pyUmbral санг ашиглахаар шийдсэн.

## 2.4 Бүлгийн дүгнэлт

Прокси дахин шифрлэлт судлаж системийг хөгжүүлэлтэнд шаардлагатай технологиуд сангуудыг судлаж хөгжүүлэлт хийж эхэлсэн.

---

## БҮЛЭГ 3

---

Прокси дахин шифрлэлтэд суурилсан  
файл хуваалцах систем хөгжүүлэх

### 3.1 Системийн шаардлага

Систем нь түлхүүр үүсгэх файл хадгалах сервер, өгөгдөл эзэмшигч, өгөгдөл хэрэглэгч гэсэн үндсэн гурван хэсгээс тогтоно. Өгөгдөл эзэмшигч бүртгэл үүсгэж өөрийн нийтийн түлхүүр болон хувийн түлхүүрийг үүсгэж авна. Файл шифрлэх болон тайлах үйлдлийг өөрийн төхөөрөмж дээр үйлдэнэ.

#### Системийн оролцогч

- Хэрэглэгч

#### Системийн тоглогч

- Файл эзэмшигч
- Файл хэрэглэгч

#### Функцийн шаардлага

Файл эзэмшигчийн функционал шаардлага:

- Системд өөрийн бүртгэлийг үүсгэх
- Файл оруулах, шифрлэх
- Файлыг тайлах хэрлэгч сонгох
- Шифрлэсэн файлыг хуваалцсан хэрлэгчдийн жагсаалт

Файл хэрэглэгчийн функционал шаардлага:

- Системд өөрийн бүртгэлийг үүсгэх
- Өөрт хуваагдсан файлын жагсаалт
- Файлыг татаж авах
- Шифрлэсэн файлыг тайлах

#### Функцийн бус шаардлага

1. Файлыг шифрлэх, шифрийг тайлах хурдан гүйцэтгэдэг байх
2. Хэрэглэгчийн интерфейс ойлгомжтой энгийн байх.
3. Прокси серверт файлыг шифрлэсэн байдлаар хадгалах, хуваалцах
4. Өөрийн бүртгэлийг ашиглаж нэвтрэх
5. Хувийн түлхүүрийг хэрлэгчийн төхөөрөмж дээр авч явах

#### Юзкейс диаграмм

Системийн хэрлэгчид ямар үйлдлүүдийг системд хийж болохыг харуулсан хэрэглээний диаграммыг харууллаа.

Бүртгүүлэх	
ID	1
Үндсэн тоглогч	Файл эзэмшигч болон файл хэрэглэгч
Тодорхойлолт	Хэрэглэгч шинээр бүртгэл үүсгэх
Өмнөх нөхцөл	Шинэ хэрлэгч байх
Үндсэн урсгал	Хэрэглэгчийн мэдээллийг авч өгөгдлийн санд шинэ хэрэглэгч үүсгэх
Дараах нөхцөл	Хэрэглэгч өөрийн бүртгэлтэй болох нэвтрэх боломжтой болох

Хүснэгт 3.1: Бүртгүүлэх юзкейсийн тодорхойлолт

Нэвтрэх	
ID	2
Үндсэн тоглогч	Файл эзэмшигч болон файл хэрэглэгч
Тодорхойлолт	Системд нэвтрэх
Өмнөх нөхцөл	Бүртгэлтэй хэрлэгч байх
Үндсэн урсгал	Хэрлэгчийн нууц үг бүртгэл таарч байгааг шалган нэвтрүүлэх
Дараах нөхцөл	Бусад хэрэглээ рүү хандах боломжтой болох

Хүснэгт 3.2: Нэвтрэх юзкейсийн тодорхойлолт

Файл шифрлэх	
ID	3
Үндсэн тоглогч	Файл эзэмшигч
Тодорхойлолт	Файл эзэмшигч файлыг шифрлэх
Өмнөх нөхцөл	Систем нэвтэрсэн байх
Үндсэн урсгал	Файлыг сонгох Санамсаргүй байдлаар түлхүүр үүсгэх Файлыг тухайн түлхүүрээр шифрлэх
Дараах нөхцөл	Тухайн төхөөрөмж дээр шифрлэгдсэн байх

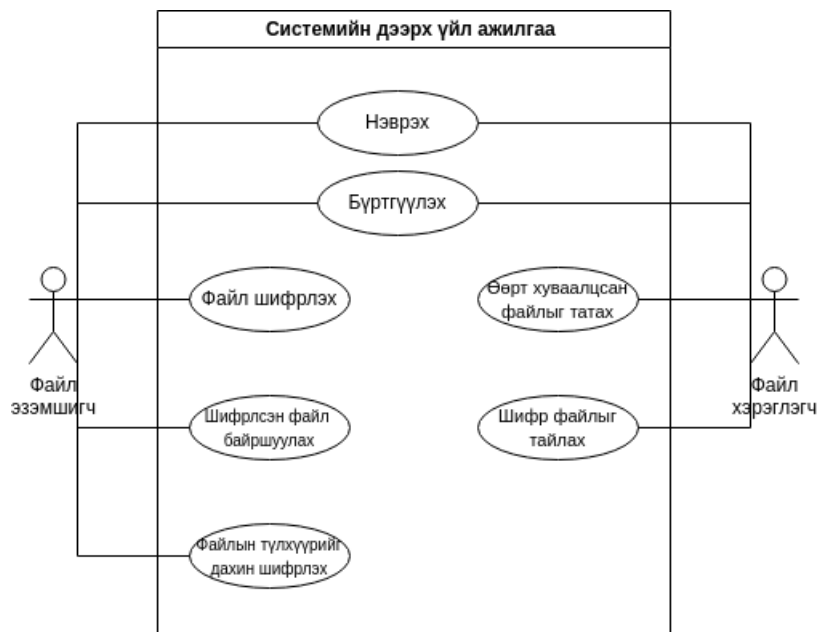
Хүснэгт 3.3: Файл шифрлэх юзкейсийн тодорхойлолт

Шифрлэсэн файл серверт байршуулах	
ID	4
Үндсэн тоглогч	Файл эзэмшигч
Тодорхойлолт	Шифрлэсэн файлыг серверт хуулна
Өмнөх нөхцөл	Файлыг шифрлэсэн байх
Үндсэн урсгал	Шифрлэсэн файлыг серверт илгээж Файлын мэдээллийг өгөгдлийн санд нэмэх
Дараах нөхцөл	Файлын талаар мэдээллийг авах боломжтой болох

Хүснэгт 3.4: Шифрлэсэн файл серверт байршуулах юзкейсийн тодорхойлолт

Файл хуваалцах	
ID	5
Үндсэн тоглогч	Файл эзэмшигч
Тодорхойлолт	Файл эзэмшигч файл хуваалцах хүнийг сонгох
Өмнөх нөхцөл	Файлыг серверт байршуулсан байх
Үндсэн урсгал	Файл хуваалцах хүний нийтийн түлхүүрийг авах Дахин шифрлэх түлхүүр үүсгэх Файлыг дахин шифрлэх
Дараах нөхцөл	Шифрлэсэн файлыг төхөөрөмж дээр тайлхда бэлэн болох

Хүснэгт 3.5: Файл хуваалцах юзкейсийн тодорхойлолт



ЗУРАГ 3.1: Юзкейс диаграмм

Файл татах	
ID	6
Үндсэн тоглогч	Файл хэрэглэгч
Тодорхойлолт	Дахин шифрлэсэн файлыг татаж авах
Өмнөх нөхцөл	Файлыг хуваалцсан байх
Үндсэн урсгал	Файлыг татаж авах
Дараах нөхцөл	Файлыг тайлахад бэлэн болох

Хүснэгт 3.6: Файл татах юзкейсийн тодорхойлолт

## Юзкейс тодорхойлолт

### 3.2 Системийн загвар

#### Үйл ажилгааны диаграмм

#### Өгөгдлийн сангийн бүтэц

### 3.3 Системийн хөгжүүлэх

Шифрлэсэн файлын тайлах	
ID	7
Үндсэн тоглогч	Файл хэрэглэгч
Тодорхойлолт	Татаж авсан шифрлэсэн файлыг тайлах
Өмнөх нөхцөл	Файлыг хуваалцсан байх
Үндсэн урсгал	Файлын түлхүүрийг өөрийн хувийн түлхүүрээр тайлах Файлын түлхүүрээр файлыг тайлах
Дараах нөхцөл	Файлыг унших боломжтой болгох

Хүснэгт 3.7: Шифрлэсэн файлын тайлах юзкейсийн тодорхойлолт

# Ном зүй

- [1] Amazon Web Services. *What Is Data Sharing?* URL: <https://aws.amazon.com/what-is/data-sharing>.
- [2] *Why data security is vital for the well-being of any enterprise today*. URL: <https://www.ibm.com/topics/data-security>.
- [3] Wikipedia. *Identity-based encryption*. URL: [https://en.wikipedia.org/wiki/Identity-based\\_encryption](https://en.wikipedia.org/wiki/Identity-based_encryption).
- [4] Wikipedia. *Homomorphic encryption*. URL: [https://en.wikipedia.org/wiki/Homomorphic\\_encryption](https://en.wikipedia.org/wiki/Homomorphic_encryption).
- [5] Wikipedia. *Encrypting File System*. URL: [https://en.wikipedia.org/wiki/Encrypting\\_File\\_System](https://en.wikipedia.org/wiki/Encrypting_File_System).
- [6] Giuseppe Ateniese **and others**. “Improved proxy re-encryption schemes with applications to secure distributed storage”. in *ACM Transactions on Information and System Security (TISSEC)*: 9.1 (2006), pages 1–32.