سوف نقوم ببناء نظام بنكي للحوالات المالية , والذي يتيح للمستخدمين إمكانية ارسال الحوالات المالية الى بعضهم البعض بطريقة أمنة .

هنا سوف يتم تقديم المرحلتين الأولى والثانية من بناء النظام كما هو مطلوب , فسوف نركز الشرح بشكل أساسي على الآلية التي تمت بها عمليات التشفير وفك التشفير في كلا المرحلتين .

في البداية يجب التعريف بالخدمات التي يقدمها هذا النظام , والتي هي :

1. **Registration**: خدمة انشاء حساب بنكي جديد ضمن هذا النظام.
2. **Log in**: خدمة تسجيل دخول الى الحساب البنكي الموجود مسبقاً.
3. **Recharge**: خدمة زيادة الرصيد المالي ضمن الحساب البنكي.
4. **Transformation**: خدمة ارسال حوالة مالية من عميل الى آخر.
5. **Show balance**: خدمة اظهار القيمة المالية الحالية ضمن الحساب البنكي.
6. **Deposits**: خدمة اظهار جميع عمليات التحويل التي قام بها العميل الحالي.

الآن سوف نقوم بشرح آلية تحقيق كل خدمة من الخدمات السابقة مع توضيح بشكل أساسي عمليات التشفير وفك التشفير بدقة في حال وجودها ضمن الخدمة المقدمة .

ملاحظة : تم الاعتماد على نموذج Server – Client Model , وتم الاتصال من خلال الاعتماد على Socket

REGISTRATION

عند تشغيل التطبيق عند المستخدم (العميل) , تظهر لهُ في البداية واجهة تتيح لهُ اما انشاء حساب بنكي جديد او تسجيل الدخول الى حساب بنكي موجود سابقاً .

فاذا لم يكن للمستخدم أي حساب بنكي واراد الحصول على حساب جديد , فيقوم المستخدم بعملية Registration , وبالتالي يتم تنفيذ الخطوات التالية :

Client

1. يرسل الى المخدم (Server) نوع الخدمة التي يريدها , في هذهِ الحالة الخدمة هي Registration .
2. يرسل الى المخدم الاسم الذي يريد ان يتم تعيينهُ على انهُ اسم مستخدم لهذا الحساب البنكي .
3. يستقبل من المخدم قيمة الidالمنشئ الخاص بهِ , بحيث لكل عميل id خاص بهِ ضمن هذا النظام.
4. ثم يقوم العميل بتوليد مفتاح Public Key خاص بهِ , ويرسلهُ الى المخدم .

Server

1. يستقبل نوع الخدمة التي يريدها العميل (Client) .
2. يستقبل الاسم الخاص بهِ .
3. يولد مفتاح تشفير , يجب على العميل ان يأخذهُ يدوياً , لأنهُ لن يتم ارسالهُ عبر الشبكة .
4. يرسل الid الجديد الخاص بالعميل المسجل اليهِ .
5. ويقرأ المفتاح Public Key الخاص بالعميل .
6. ويتم إضافة معلومات هذا العميل الى قاعدة البيانات الخاصة بالنظام ,(id , Public Key) .

ويتم تكرار هذه الخطوات في كل عملية انشاء حساب بنكي جديد على هذا النظام من قِبل المستخدم .

LOG IN

اذا كان العميل يملك حساب بنكي سابق على هذا النظام , يمكنهُ تسجيل الدخول اليهِ فقط من خلال ادخال رقم الid الخاص بهِ , والذي تم الحصول عليهِ من خلال عملية انشاء هذا الحساب , وبالتالي يتم تنفيذ الخطوات التالية :

Client

1. يرسل الى المخدم (Server) نوع الخدمة التي يريدها , في هذهِ الحالة الخدمة هي Log in .
2. يرسل الid الخاص بهِ الى المخدم .
3. يستقبل ويقرأ الid الذي يرسلهُ المخدم من اجل عملية التحقق من صحة وجود الid المدخل من قبل العميل (ففي حال كان الid صحيح وموجود ضمن قاعدة البيانات يتم ارسال نفس ال id , والا سوف يتم ارسال 0-1) .

Server

1. يستقبل نوع الخدمة التي يريدها العميل (Client) .
2. يستقبل ويقرأ الid الخاص بالعميل .
3. يتحقق من وجود الid المُرسل ضمن قاعدة البيانات لديهِ , ويعيد ويرسل النتيجة الى العميل .

ويتم تكرار هذه الخطوات في كل عملية تسجيل الدخول الى حساب بنكي على هذا النظام من قِبل المستخدم .

RECHARGE

الآن لنفترض ان العميل داخل حسابهُ البنكي , ويريد ان يضع المزيد من المال ضمن حسابهُ , فيجب عليهِ ان يضغط على خيار **Recharge** , وان يضع القيمة المالية التي يريد اضافتها , بالتالي يتم تنفيذ الخطوات التالية :

Client

1. يرسل الى المخدم (Server) نوع الخدمة التي يريدها , في هذهِ الحالة الخدمة هي recharge.
2. يرسل الid الخاص بهِ الى المخدم .
3. يرسل نوع التشفير الذي يريد ان يتم بهِ حماية المعلومات المتبادلة بينهُ وبين المخدم
   (1 : تشفير متناظر , 2 : PGP) .
4. في حال كان التشفير متناظر (Symmetric) :
   1) يقوم بتوليد (initializing Vector) iv وذلك لأننا نستخدم (CBC Mode) .
   2) ثم يقوم بتشفير القيمة المالية التي يريد ان يضيفها الى حسابه يقومُ , ويرسلها الى المخدم .
   3) يستقبل ويقرأ الرد من المخدم , بحيث سوف يكون الرد مشفراً بنفس الطريقة , فيجب فك تشفيرهُ , ومعرفة انهُ تمت العملية بنجاح .
5. في حال كان التشفير هجين (PGP) :
   1) يقوم بتوليد مفتاح الجلسة التناظري , ويشفرهُ باستخدام المفتاح العام الخاص بالمخدم (Public Key Server) , ويرسلهُ الى المخدم .
   2) ثم يقوم بتشفير القيمة المالية باستخدام مفتاح الجلسة التناظري , ويرسلها الى المخدم .
   3) يستقبل ويقرأ الرد من المخدم , بحيث سوف يكون الرد مشفراً بنفس الطريقة , فيجب فك تشفيرهُ , ومعرفة انهُ تمت العملية بنجاح .

Server

1. يستقبل نوع الخدمة التي يريدها العميل (Client) .
2. يستقبل ويقرأ الid الخاص بالعميل .
3. يستقبل ويقرأ نوع التشفير الذي يريدهُ العميل .
4. في حال كان التشفير متناظر (Symmetric) :
   1) يستقبل ويقرأ iv (initializing Vector) .
   2) يستقبل ويقرأ القيمة المالية المشفرة .
   3) يتم الحصول على المفتاح التناظري الخاص بهذا العميل من قاعدة البيانات من خلال ربطهُ مع الid الخاص بهِ.
   4) يتم فك تشفير القيمة المالية بالاعتماد على المفتاح التناظري .
   5) ومن ثم يتم إضافة المبلغ المالي المرسل الى الرصيد المالي الموجود ضمن هذا الحساب .
   6) ويقوم بتشفير رسالة تأكيد إتمام العملية بنجاح , وارسالها الى العميل ليأكد لهُ بتمام العملية بنجاح .
5. في حال كان التشفير هجين (PGP) :
   1) يقوم باستقبال مفتاح الجلسة التناظري المشفر باستخدام المفتاح العام (Public Key Server) , ويفك تشفيرهُ باستخدام المفتاح الخاص (Private Key Server) .
   2) يستقبل ويقرأ القيمة المالية المشفرة .
   3) يقوم بفك تشفير القيمة المالية باستخدام مفتاح الجلسة التناظري .
   4) ومن ثم يتم إضافة المبلغ المالي المرسل الى الرصيد المالي الموجود ضمن هذا الحساب .
   5) ويقوم بتشفير رسالة تأكيد إتمام العملية بنجاح , وارسالها الى العميل ليأكد لهُ بتمام العملية بنجاح .

ويتم تكرار هذه الخطوات في كل عملية تعبئة رصيد مالي ضمن الحساب بنكي على هذا النظام من قِبل المستخدم .

TRANSFORMATION

الآن يريد العميل ان يرسل حوالة مالية الى عميل آخر من خلال هذا النظام البنكي , بالتالي يتم تنفيذ الخطوات التالية :

Client

1. يرسل الى المخدم (Server) نوع الخدمة التي يريدها , في هذهِ الحالة الخدمة هي transformation.
2. يرسل الid الخاص بهِ الى المخدم .
3. يرسل نوع التشفير الذي يريد ان يتم بهِ حماية المعلومات المتبادلة بينهُ وبين المخدم
   (1 : تشفير متناظر , 2 : PGP) .
4. في حال كان التشفير متناظر (Symmetric) :
   1) يقوم بتوليد (initializing Vector) iv وذلك لأننا نستخدم (CBC Mode) .
   2) ثم يقوم بتشفير القيمة المالية التي يريد ان يحولها الى العميل الآخر , ويرسلها الى المخدم .
   3) ثم يقوم بتشفير قيمة الid الخاص بالعميل المراد تحويل المال الى حسابهُ , ويرسلها الى المخدم .
   4) ثم يقوم بتشفير سبب التحويل , ويرسله الى المخدم .
   4) يستقبل ويقرأ الرد من المخدم , بحيث سوف يكون الرد مشفراً بنفس الطريقة , فيجب فك تشفيرهُ , ومعرفة انهُ تمت العملية بنجاح ام لاء (حالة الحوالة المالية اكبر من الرصيد ضمن الحساب) .
6. في حال كان التشفير هجين (PGP) :
   1) يقوم بتوليد مفتاح الجلسة التناظري , ويشفرهُ باستخدام المفتاح العام الخاص بالمخدم (Public Key Server) , ويرسلهُ الى المخدم .
   2) ثم يقوم بتشفير القيمة المالية باستخدام مفتاح الجلسة التناظري , ويرسلها الى المخدم .
   3) ثم يقوم بتشفير قيمة الid الخاص بالعميل المراد تحويل المال الى حسابهُ , ويرسلها الى المخدم .
   4) ثم يقوم بتشفير سبب التحويل , ويرسله الى المخدم .
   5) يستقبل ويقرأ الرد من المخدم , بحيث سوف يكون الرد مشفراً بنفس الطريقة , فيجب فك تشفيرهُ , ومعرفة انهُ تمت العملية بنجاح .

Server

1. يستقبل نوع الخدمة التي يريدها العميل (Client) .
2. يستقبل ويقرأ الid الخاص بالعميل المُرسل .
3. يستقبل ويقرأ نوع التشفير الذي يريدهُ العميل .
4. في حال كان التشفير متناظر (Symmetric) :
   1) يستقبل ويقرأ iv (initializing Vector) .
   2) يستقبل ويقرأ القيمة المالية المشفرة .
   3) يتم الحصول على المفتاح التناظري الخاص بهذا العميل من قاعدة البيانات من خلال ربطهُ مع الid الخاص بهِ.
   4) يتم فك تشفير القيمة المالية بالاعتماد على المفتاح التناظري .
   5) يستقبل ويقرأ قيمة الid المشفرة الخاص بالعميل المراد تحويل المال الى حسابهُ, ويتم فك تشفيرها بالاعتماد على المفتاح التناظري .
   6) يستقبل ويقرأ سبب التحويل مشفر , ويتم فك تشفيره بالاعتماد على المفتاح التناظري .
   7) ومن ثم يتم التحقق في حال إمكانية إتمام عملية التحويل (أي يوجد رصيد كافي) ام لا .
   8) ففي حال وجود رصيد كافي يتم طرح قيمة المبلغ المالي المحول من رصيد المرسل , واضافتهُ الى رصيد المستقبل , وفي حال لم يوجد رصيد كافي لا يتم اجراء أي تعديل .
   9) ويقوم بتشفير رسالة تأكيد إتمام العملية (نجاح او فشل) , وارسالها الى العميل.
5. في حال كان التشفير هجين (PGP) :
   1) يقوم باستقبال مفتاح الجلسة التناظري المشفر باستخدام المفتاح العام (Public Key Server) , ويفك تشفيرهُ باستخدام المفتاح الخاص (Private Key Server) .
   2) يستقبل ويقرأ القيمة المالية المشفرة , ويقوم بفك تشفير القيمة المالية باستخدام مفتاح الجلسة التناظري .
   3) يستقبل ويقرأ قيمة الid المشفرة الخاص بالعميل المراد تحويل المال الى حسابهُ, ويتم فك تشفيرها بالاعتماد على مفتاح الجلسة التناظري .
   4) يستقبل ويقرأ سبب التحويل مشفر , ويتم فك تشفيره بالاعتماد على مفتاح الجلسة التناظري .
   5) ومن ثم يتم التحقق في حال إمكانية إتمام عملية التحويل (أي يوجد رصيد كافي) ام لا .
   6) ففي حال وجود رصيد كافي يتم طرح قيمة المبلغ المالي المحول من رصيد المرسل , واضافتهُ الى رصيد المستقبل , وفي حال لم يوجد رصيد كافي لا يتم اجراء أي تعديل .
   7) ويقوم بتشفير رسالة تأكيد إتمام العملية (نجاح او فشل) , وارسالها الى العميل.

ويتم تكرار هذه الخطوات في كل عملية ارسال حوالة مالية من عميل الى آخر .

SHOW BALANCE

يريد العميل الاستعلام عن رصيدهُ المالي ضمن البنك , يتم تنفيذ الخطوات التالية :

Client

1. يرسل الى المخدم (Server) نوع الخدمة التي يريدها , في هذهِ الحالة الخدمة هي show balance.
2. يرسل الid الخاص بهِ الى المخدم .
3. يرسل نوع التشفير الذي يريد ان يتم بهِ حماية المعلومات المتبادلة بينهُ وبين المخدم
   (1 : تشفير متناظر , 2 : PGP) .
4. في حال كان التشفير متناظر (Symmetric) :
   1) يستقبل ويقرأ iv (initializing Vector) .
   2) يستقبل ويقرأ القيمة المالية مشفرة , ويقوم بفك تشفيرها من خلال المفتاح التناظري الخاص بهِ .
5. في حال كان التشفير هجين (PGP) :
   1) يقوم بتوليد مفتاح الجلسة التناظري , ويشفرهُ باستخدام المفتاح العام الخاص بالمخدم (Public Key Server) , ويرسلهُ الى المخدم .
   2) يستقبل ويقرأ القيمة المالية مشفرة , ويقوم بفك تشفيرها من خلال مفتاح الجلسة التناظري.

Server

1. يستقبل نوع الخدمة التي يريدها العميل (Client) .
2. يستقبل ويقرأ الid الخاص بالعميل .
3. يستقبل ويقرأ نوع التشفير الذي يريدهُ العميل .
4. في حال كان التشفير متناظر (Symmetric) :
   1) يقوم بالحصول على المفتاح التناظري الخاص بالعميل من خلال ربط الid الخاص بهِ مع قيمة المفتاح ضمن قاعدة البيانات .
   2) يقوم بتوليد iv (initializing Vector) .
   3) يقوم بتشفير قيمة الرصيد المالي للعميل باستخدام المفتاح الخاص بهِ , والiv , ويرسلهُ للعميل .
5. في حال كان التشفير هجين (PGP) :
   1) يقوم باستقبال مفتاح الجلسة التناظري المشفر باستخدام المفتاح العام (Public Key Server) , ويفك تشفيرهُ باستخدام المفتاح الخاص (Private Key Server) .
   2) يقوم بتشفير قيمة الرصيد المالي للعميل باستخدام مفتاح الجلسة التناظري , ويرسلهُ للعميل .

DEPOSITS

الآن اخيراً يريد العميل اظهار جميع عمليات التحويل التي قام بها من حسابهُ الخاص والتي أتت الى حسابهُ :

Client

1. يرسل الى المخدم (Server) نوع الخدمة التي يريدها , في هذهِ الحالة الخدمة هي deposits.
2. يرسل الid الخاص بهِ الى المخدم .
3. يرسل نوع التشفير الذي يريد ان يتم بهِ حماية المعلومات المتبادلة بينهُ وبين المخدم
   (1 : تشفير متناظر , 2 : PGP) .
4. في حال كان التشفير متناظر (Symmetric) :
   1) يستقبل ويقرأ iv (initializing Vector) .
   2) يستقبل ويقرأ عمليات التحويل مشفرة , ويقوم بفك تشفيرها من خلال المفتاح التناظري الخاص بهِ .
5. في حال كان التشفير هجين (PGP) :
   1) يقوم بتوليد مفتاح الجلسة التناظري , ويشفرهُ باستخدام المفتاح العام الخاص بالمخدم (Public Key Server) , ويرسلهُ الى المخدم .
   2) يستقبل ويقرأ عمليات التحويل مشفرة , ويقوم بفك تشفيرها من خلال مفتاح الجلسة التناظري.

Server

1. يستقبل نوع الخدمة التي يريدها العميل (Client) .
2. يستقبل ويقرأ الid الخاص بالعميل .
3. يقوم بالحصول على جميع عمليات التحويل الخاصة بهذا العميل وذلك عن طريق ربط الid الخص بهِ ضمن قاعدة البيانات .
4. يستقبل ويقرأ نوع التشفير الذي يريدهُ العميل .
5. في حال كان التشفير متناظر (Symmetric) :
   1) يقوم بالحصول على المفتاح التناظري الخاص بالعميل من خلال ربط الid الخاص بهِ مع قيمة المفتاح ضمن قاعدة البيانات .
   2) يقوم بتوليد iv (initializing Vector) .
   3) يقوم بتشفير عمليات التحويل للعميل باستخدام المفتاح الخاص بهِ , والiv , ويرسلهُ للعميل .

6. في حال كان التشفير هجين (PGP) :

1) يقوم باستقبال مفتاح الجلسة التناظري المشفر باستخدام المفتاح العام (Public Key Server) , ويفك تشفيرهُ باستخدام المفتاح الخاص (Private Key Server) .
2) يقوم بتشفير قيمة عمليات التحويل للعميل باستخدام مفتاح الجلسة التناظري , ويرسلهُ للعميل .

ويتم تكرار هذه الخطوات في كل عملية Deposits .

وبهذا الشكل قد تم توضيح آلية عمل النظام , وجميع آليات التشفير التي تؤمن الحماية على الاتصال بين العملاء والمخدم ضمن هذا النظام .

THE END

-.-