Таблица 3

Генератор	Количество тестов, в которых тестирование прошло больше 99% последовательностей	Количество тестов, в которых тестирование прошло больше 96% последовательностей	Количество, тестов в которых значение вероятности		
			p<=0.01	p<= 0.001	p<=0.05
RSB	130	187	0	0	6
ГОСТ 28147-98	130	186	1	0	3
DES	125	188	1	0	16
IDEA	125	187	1	0	7
AES	125	188	2	0	8

В качестве открытого текста для всех генераторов был выбран один и тот же 17-Мбайтный русскоязычный текст (словарь Даля).

Выводы

1. Шифр RSB имеет четкую и ясную структуру, за один раунд в нем преобразуются все биты входного блока, в отличие от шифра Фейстеля (являющегося основой большинства криптоалгоритмов), где за один раунд изменяется, как правило, лишь половина бит входного блока.

2. Как показали результаты статистических испытаний ***RSB*** криптосистемы для 256 – разрядного ключа ($r = 8$) и 32–х итераций шифрования, эффективность алгоритма зашифрования, оцениваемая количеством тестов в пакете ***NIST STS***, в котором тестирование прошло больше 99% и соответственно 96 % последовательностей, оказалось на уровне российского алгоритма ***ГОСТ*** 28147-89 и превосходит эффективность широко используемых стандартов криптографической защиты, таких как ***DES***, ***IDEA*** и ***AES*** (*Rijndael*).

Список литературы

1.Мао В. Современная криптография. Теория и практика. – М.: «Вильямс», 2005. – 768с.

2.Шнайер Б. Прикладная криптография. –М.: «ТРИУМФ», 2003. – 816 с.

3.Белецкий А.Я. Комбинаторика кодов Грея. – Киев: КВІЦ, 2003. – 506 с.

4.Харин Ю.С. Берник В.И., Матвеев Г.В., Агиевич С.В. Математические и компьютерные основы криптологии. – Мн.: Новое знание, 2003. – 382 с.

5.Молдавян Н.А., Молдавян А.А., Еремееев М.А. Криптография: от примитивов к синтезу алгоритмов. – СПб.: БХВ, 2004. – 448 с.

6.Random Number Generation and Testing. http: www.csrc.nist.gov/rng/

УДК 681.04

О.С. Петров, Є.О.Валуйський

ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ІНФОРМАЦІЇ В ПРОЦЕСІ ДИСТАНЦІЙНОГО НАВЧАННЯ НА ОСНОВІ ТЕХНОЛОГІЇ СМАРТ-КАРТ.

Вступ

Для забезпечення більш ефективного захисту інформації, аутентифікації користувачів, цілісного контролювання процесу дистанційного навчання, потрібно розробити адекватну модель верифікації. Необхідно забезпечити аутентичнісить користувача, на рівні єдиного документу (студентський, читацький квиток) для доступу до електронних ресурсів (бібліотека, комплект учбово-методичної документації тощо), послуг Інтернет, а також забезпечення цілісності інформації в процесі передачі й прийому даних через відкриті мережі при тестуванні студента в процесі дистанційного навчання.

У цей час, більша частина сучасних технологій проведення тестування має один вагомий недолік. Всі вони хоча й допускають деяку аутентифікацію користувача, але, не дозволяють повною мірою проконтролювати цей процес. Недбалість у даному питанні знижує цінність результату тесту, а також, частіше всього, приводить до наступних проблем:

1. Дозволяє учасникам попередньо ознайомитися зі списком питань і варіантами відповідей.
2. За допомогою аналізатора трафіка дає можливість учасникам визначити відповіді інших.
3. Дозволяє скористатися некоректними реєстраційними даними.
4. Не виключає можливість підміни учасника тестування.

Незважаючи на порівняно невелику ймовірність використання кожного пункту, окремо, їхня сукупність зменшує рівень довіри до тестів на основі сучасних комп'ютерних технологій. У деяких же випадках, їхнє застосування взагалі стає недоцільним.

Технологія SmartCard

Для усунення вищеописаних недоліків пропонується використати технологію SmartCard (Смарт-карт).

Смарт-карти нараховують цілу низку переваг[1]. Маючи вигляд звичайної пластикової кредитної картки, вона містить у собі інтегральну схему, що наділяє її здатностями до зберігання й обробки інформації.

Смарт-карта, по своїм розмірам схожа на кредитну, здатна зберігати й обробляти інформацію за допомогою напівпровідникових електронних схем, захованих усередині пластикового корпусу.

Залежно від убудованої мікросхеми всі смарт-карти поділяються на кілька основних типів, що кардинально розрізняються по виконуваним функціям:

- карти пам'яті;
- мікропроцесорні карти;
- карти із криптографічною логікою.

Карти пам'яті призначені для зберігання інформації. Пам'ять на таких типах карт може бути вільною для доступу або містити логіку контролю доступу до пам'яті карти для обмеження операцій читання й запису даних.

Мікропроцесорні карти також призначені для зберігання інформації, але у відмінність від звичайних карт пам'яті вони містять у собі спеціальну програму або невелику операційну систему, що дозволяє перетворювати дані по певному алгоритму, здійснювати захист інформації збереженої на карті при передачі, читанні й запису, виконувати криптографічні перетворення інформації (схеми шифрування, електронного цифрового підпису, аутентифікації та ін.). Крім того, ці смарт-карти можуть додатково забезпечуватися засобами аутентифікації (як логічного, так і біометричного характеру), власника смарт-карти.

Карти із криптографічною логікою використовуються в системах захисту інформації для прийняття особистої участі в процесі шифрування даних або вироблення криптографічних ключів, електронних цифрових підписів і іншої необхідної інформації для роботи системи.

Зараз смарт-карти з мікропроцесором одержують все більше поширення в усьому світі завдяки властивим їм широких можливостям по аутентифікації користувачів таких карт, компактності використання, швидкості й зручності роботи з ними, а також стійкості застосованих в них криптографічних алгоритмів, що забезпечує високу конфіденційність інформації. Доступ до даних, збережених на карті, контролюється мікропроцесором карти за допомогою пароля (ПИН).

Для реалізації програмно-апаратного комплексу становить інтерес технологія Java Card[2]. Дана технологія використовує смарт-карти, що підтримують програми, написані мовою Java.

Структура Java Card визначає набір класів прикладного програмного інтерфейсу (API (Application Program Interface)), призначеного для розробки додатків Java Card, і набір системних служб для нормального функціонування цих додатків. Необхідність додаткових бібліотек, які реалізують розширені службові функції, що забезпечують безпеку й описують системну модель, визначається специфікою розв'язуваних завдань. Додатки Java Card називаються апплетами. На одній карті може виконуватися декілька різних апплетів. Кожний апплет визначається ідентифікатором додатка (application identifier, AID), що описується в п'ятій частині стандарту ISO 7816. Апплет не просто програє той самий сценарій, а реагує на дії користувача й може динамічно міняти своє поводження.

Один із ключових принципів розробки мови Java полягав у забезпеченні захисту від несанкціонованого доступу. Програми на Java не можуть викликати глобальні функції й одержувати доступ до довільних системних ресурсів, що забезпечує в Java рівень безпеки, недоступний для інших мов.

Використання мови Java дає наступні переваги:

- Java надає для широкого використання свої апплети (applets) - невеликі, надійні, динамічні, що не залежать від платформи активні мережеві додатки, що вбудовують у сторінки Web. Апплети Java можуть настроюватися й поширюватися споживачам з такою ж легкістю, як будь-які документи HTML.
- Java вивільняє міць объектно-орієнтованної розробки додатків, сполучаючи простий і знайомий синтаксис із надійним і зручним в роботі середовищем розробки. Це дозволяє широкому колу програмістів швидко створювати нові програми й нові апплети.

Програмні та апаратні вимоги

Для реалізації пропонованого комплексу, сервер повинен відповідати низці вимог. Проведений аналіз розглянутої проблеми дозволив сформулювати вимоги до програмного забезпечення серверної частини:

- ОС: Linux (версія ядра старше 2.4) або FreeBSD;
- Web-сервер: Apache 1.3 і вище з модулями openssl і PHP старше 4.1;
- БД: MySQL 4 чи пізніша;
- Програмний або апаратний файервол.

Апаратна частка сервера залежить від обраної операційної системи й середнього навантаження на сервер. Крім цього, бажано доповнити сервер пристроєм резервного копіювання й оснастити сервер апаратним генератором випадкових чисел.

Для роботи клієнтської частини необхідно оснастити комп'ютери кард-рідерами, а також повинне бути встановлене відповідне програмне забезпечення - браузер з підтримкою JavaScript 1.2 і віртуальна машина Java від Sun.

У якості кард-рідера можна використати зовнішній пристрій читання/запису контактних смарт-карт ACR30U-CFC. Даний карт-рідер являє собою компактний і зручний пристрій, що підтримує всі мікропроцесорні смарт-карти, а також більшість карт пам'яті всіх відомих виробників.

Виходячи з існуючих завдань по захисту й вірогідності інформації, що виникають у процесі навчання, пропонується наступний алгоритм роботи комплексу:

1. Кожний користувач отримує на руки персональну смарт-карту. Карта містить логін, пароль і ключ користувача. Технологія знижує до мінімуму можливість неправомірного завантаження карти з метою заміни цих даних. Можливо, також об'єднання такої смарт-карти з існуючими студентськими й читацькими квитками. Це дозволить найбільш широко впровадити технологію в існуючий навчальний процес.
2. Перед початком тесту користувач вставляє свою смарт-карту в рідер. Клієнтська частина додатка визначає наявність смарт-карти в рідері й зчитує відтіля реєстраційні дані. Виконується базова перевірка цих даних (цілісність і т.п.), після чого додаток установлює захищене з'єднання з сервером. Клієнт відправляє серверу реєстраційні

дані, зчитані зі смарт-карти. У свою чергу, сервер перевіряє наявність у базі даних користувачів подібного запису. З метою підвищення захищеності системи паролі зберігаються в базі у вигляді криптографічних відбитків.

3. Якщо подібний запис у базі знайдено, сервер виконує також ряд додаткових перевірок. При проведенні тестування, сервер перевіряє наявність дозволу на здачу тесту в поточний проміжок часу (щоб уникнути несанкціонованої перездачі тесту іншим часом).
4. Якщо користувача авторизовано вдало, сервер передає клієнтові необхідні дані. Використання захищеного з'єднання зводить нанівець можливість перехоплення даних.
5. Апплет стежить за наявністю карти в рідері. У випадку витягу карти з рідера з'єднання клієнта із сервером розривається. У свою чергу, при проведенні тестування, сервер стежить за тим, щоб дані користувача були відіслані у відведений для тесту проміжок часу. По закінченні даного часу сервер також розриває з'єднання. Коли тестування закінчено відповіді користувача підписуються його персональним ключем зі смарт-карти й зберігаються в БД сервера. Таким чином, викладач може перевірити цілісність результатів тестування для кожного студента. Це унеможливлює несанкціоновану вставку в БД відповідей і результатів тесту.

Розглядаючи інформаційну взаємодію студента й університету з погляду безпеки, варто якісно вирішити кілька завдань [3]:

- захист підключених до публічних каналів зв'язку локальних мереж і окремих комп'ютерів від несанкціонованих дій;
- захист інформації (особистих даних, контрольних завдань, результатів тестування) у процесі передачі по відкритих каналах зв'язку.

Для рішення першого завдання доцільне використання мережевих екранів (брандмауерів). На сьогоднішній день застосування брандмауерів є стандартним не тільки в корпоративному секторі, але і як захист окремого вилученого комп'ютера.

Рішення другого завдання припускає виконання наступних функцій [4]:

- аутентифікація взаємодіючих сторін;
- криптографічне закриття переданих даних;
- підтвердження дійсності й цілісності доставленої інформації;
- захист від повтору, затримки й видалення повідомлень;
- захисту від заперечення фактів відправлення й прийому повідомлень.

Дані функції зв'язані один з одним, їхня реалізація заснована на криптографічному захисті переданих даних. Ефективність такого захисту забезпечується за рахунок спільного використання симетричних і асиметричних криптографічних систем.

Така технологія об'єднання локальних мереж і окремих комп'ютерів через відкрите зовнішнє середовище в єдину віртуальну корпоративну мережу, що забезпечує безпеку циркулюючих у ній даних називається VPN (віртуальна захищена мережа) [5]. Для незалежності від прикладних протоколів і додатків VPN формуються на одному з більш низьких рівнів моделі OSI (Open System Interconnection) -канальному, мережному або сеансовому. Чим нижче рівень, на якому реалізується захист, тим вона прозоріше для додатка й користувача. Більш високий рівень моделі OSI розширює набір послуг безпеки і полегшує конфігурування системи захисту, але в цьому випадку підсилюється залежність від використовуваних протоколів обміну й додатків.

Розглянемо докладніше захищене з'єднання (сеансовий рівень), на якому найбільшу популярність для шифрування інформації одержав протокол SSL/TLS (Secure Sockets Layer/Transport Layer Security). Ядром цього протоколу є технологія комплексного використання симетричних і асиметричних криптосистем компанії RSA Data Security. Для аутентифікації взаємодіючих сторін і криптозахисту ключа симетричного шифрування використовується цифрові сертифікати відкритих ключів користувача (у нашому випадку, віддаленого студента й сервера дистанційного навчання), завірені цифровими підписами

спеціальних Сертифікаційних Центрів. Сертифікати відповідають загальноприйнятому стандарту X.509. Як алгоритми асиметричного шифрування використаються алгоритми RSA, а також Діффі-Хелмана. Припустимими алгоритмами симетричного шифрування є RC2, RC4, DES а також Triple DES. Для обчислення хеш-функцій можуть застосовуватись MD-5 і SHA-1. Як альтернатива існуючої й уже реалізованим у протоколі алгоритмам потокового симетричного шифрування нами намічено застосування власного потокового шифру.

Для стандартизації аутентифіцированого проходу через брандмауер консорціум IETF(Проблемна група проектування Інтернет) визначив протокол за назвою SOCKS, у цей час 5 версія цього протоколу застосовується для стандартизованої реалізації посередників каналів. У протоколі SOCKS v.5 клієнтський комп'ютер установлює аутентифіцированій сокет (або сеанс) із сервером виконуючи роль посередника (proxy). Цей посередник - єдиний спосіб зв'язку через між мережевий екран. Посередник у свою чергу, проводить будь-які операції, запитувані клієнтом.

Як платформа для розробки клієнтської частини програмного забезпечення нами обрана технологія JAVA CARD. Вона дозволяє розробляти як апплети, які виконуються внутрішнім процесором смарт-карти так і хост додаток, що підтримує обмін інформацією зі смарт-картой, мовою Java. Для роботи із протоколами SSL\TLS в Java починаючи з версії 1.4 як базове розширення використовується Java Secure Socket Extension (JSSE). Використовуючи JSSE, можна встановити безпечний обмін даними між клієнтом і сервером, керуючим кожним із прикладних протоколів, типу HTTP, Telnet або FTP. .Резюмуючи складні криптографічні алгоритми безпеки й механізми "підтвердження зв'язку", JSSE мінімізує ризик створення тонкої, але небезпечної уразливості безпеки. Крім того, спрощується прикладна розробка. Виступаючи як стандартний блок, що розроблювачі можуть безпосередньо використати у своїх додатках.

Як переваги які дає використання JSSE можна виділити наступне:

- 100% реалізований на чистій Java
- Може експортуватися в більшість країн
- Забезпечує підтримку програмного інтерфейсу додатків SSL версіям 2.0 і 3.0 і виконання SSL 3.0
- Забезпечує підтримку програмного інтерфейсу додатків і виконання для версії TLS 1.0
- Забезпечує підтримку Протоколу Передачі Гіпертексту (HTTP), інкапсульованого в протоколі SSL (HTTPS), що надає безпечний доступ до даних, веб-сторінкам, використовуючи HTTPS
- Забезпечує підтримку найпоширеніших криптографічних алгоритмів, включаючи перераховані в табл. 1.

Таблиця 1. Криптографічні алгоритми, використані для Java Secure Socket Extension

Криптографічний алгоритм	**Довжина ключа (Bits)**
RSA	2048 (authentication), 2048 (key exchange), 512(key exchange)
RC4	128, 128 (40 effective)
DES	64 (56 effective), 64 (40 effective)
Triple DES	192, (112 effective)
AES	256, 128
Diffie-Hellman	1024, 512
DSA	1024

Вищенаведений алгоритм дозволяє побудувати систему, позбавлену переважної більшості недоліків, властивим сучасним технологіям, застосованих у вищих навчальних закладах. Вона виключає підробку й зміну, як з боку студента, так і з боку технічного персоналу, що перевіряє. Подібна технологія відмінно масштабується й дозволяє побудувати централізовану систему в рамках факультету й університету. У той же час, система відкрита для доробки, і може бути доповнена іншими можливостями, наприклад, статистичним аналізом, роздруківкою звітів, відомостей тощо.

Висновки

Використання смарт-карт для захисту інформації в процесі дистанційного навчання надає наступні переваги:

- можливість автоматизовано управляти й контролювати комп'ютерну мережу університету;
- надавати послуги Інтернет студентам і технічному персоналу;
- універсальність системи дозволяє з'єднувати кілька комп'ютерних мереж у єдину керовану структуру з одним центральним сервером;
- гнучке розмежування доступу до мережних ресурсів для операторів, адміністраторів та ін.;
- одночасна робота безлічі користувачів у системі за допомогою web-інтерфейсу;
- використання мікропроцесорних смарт-карт у якості студентського й читацького квитка гарантує, що облікові записи використаються лише тими людьми, кому їх надали. Смарт-карти захищені від підробки, їх неможливо скопіювати, тому що після персоналізації карти, ключі доступу ніколи й ніде не з'являються у відкритому виді;
- користувачі унікально ідентифікують і аутентифікують себе з використанням унікального імені користувача й пароля.
- застосування смарт-карт в університеті багато в чому спрощує роботу системного адміністратора й користувача.

Список літератури

1. ISBC – Смарт-карти. http://www.isbc.ru/products/smart/

2. Sun Microsystem. Java Technology. http://java.sun.com/

3. Столлингс, Вільям. Криптографія й захист мереж: принципи й практика, 2-і изд.-М.:"Вільямс",2001.-672 с.

4. Зима В.М., Молдовян А.А., Молдовян Н.А. Безпека глобальних мережних технологій. -Спб.: Бхв-петербург, 2000.-320 с.

5.Галицкий А.В.,Рябко С.Д.,Шаньгин В.Ф. Захист інформації в мережі - аналіз технологій і синтез рішень-м.:ДМК Пресс, 2004.-616 с.