Міністерство освіти і науки України
Харківський національний університет ім. В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра безпеки інформаційних систем і технологій

КУРСОВА РОБОТА

з дисципліни «Прикладна криптологія»

на тему «Аналіз вразливостей криптографічних перетворень , що реалізується на основі перетворень в кільці.»

Виконав:
студент групи КБ-31
Кравченко Є.М.
Перевірив:
професор
Горбенко І.Д.

Харків 2020

ЗМІСТ

ВСТУП

Криптология досить чітко ділиться на дві частини: криптографію (шифрування) і криптоаналіз. Криптограф намагається знайти методи забезпечення секретності і (або) автентичності (справжності) повідомлень. Криптоаналітика намагається виконати зворотну задачу, розкриваючи шифр або підробляючи кодовані сигнали таким чином, щоб вони були прийняті як справжні. Оригінал тексту, з яким криптограф застосовує своє мистецтво, називається відкритим текстом, а результат його роботи - шифрованих текстом повідомлення - шифртекст, або криптограмою. Для управління процесом шифрування криптограф завжди використовує секретний ключ. Часто (але не завжди) він передає цей секретний ключ будь-яким надійним способом (наприклад, в "дипломаті", пристебнутий наручниками до руки кур'єра) людині (або машині), якому він збирається пізніше послати криптограму, складену з використанням цього ключа.

Майже загальноприйняте припущення в криптографії полягає в тому, що криптоаналитик противника має повний текст криптограми. Крім того, криптограф майже завжди керується правилом, вперше сформульованим голландцем Керкхоффом: стійкість шифру повинна визначатися тільки секретністю ключа. Іншими словами, правило Керкхоффом полягає в тому, що весь механізм шифрування, крім значення секретного ключа, відомий криптоаналітика противника. Якщо криптограф приймає тільки ці два припущення, то він розробляє систему, стійку при аналізі на основі тільки шифрованого тексту. Якщо до того ж криптограф допускає, що криптоаналитик противника зможе дістати кілька уривків відкритого тексту і відповідного йому шифрованого тексту, утвореного з використанням секретного ключа, то розробляється система, стійка при аналізі на основі відкритого тексту. Криптограф може навіть допустити, що криптоаналитик противника здатний ввести свій відкритий текст і отримати правильну криптограму, утворену з використанням секретного ключа (аналіз на основі

обраного відкритого тексту), або припустити, що криптоаналитик противника може підставити фіктивні криптограми і отримати текст, в який вони перетворюються при розшифрування (аналіз на основі обраного шифртекста), або допустити обидві ці можливості (аналіз на основі обраного тексту). Розробники більшості сучасних шифрів забезпечують їх стійкість до аналізу на основі обраного відкритого тексту навіть в тому випадку, коли передбачається, що криптоаналитик противника зможе вдатися до аналізу на основі шифртекста.

РОЗДІЛ 1

1.1 Завдання криптографії. Поняття стійкості криптографічного алгоритму.

Криптология - "особлива" область досліджень. Про досягнення цієї науки все частіше повідомляють не тільки наукові, але й науково-популярні журнали і звичайна преса. За кордоном в останні роки спостерігається небувалий бум в області криптології. Це пов'язано з тим, що її досягнення стали застосовуватися не тільки у вузьких відомчих колах, а й у житті мільйонів громадян. Широке впровадження обчислювальних систем привело до того, що вони стає привабливими для різного роду інформаційних нападів. Це полегшується тим, що інформація виявилася позбавленою свого фізичного втілення, як було раніше (наприклад, текст написаний на папері і підписаний автором). Відсутність такого фізичного втілення, поєднане з неможливістю аутентифікації його автора, відкрило шлях до різного роду порушень. У зв'язку з цим виникла необхідність не тільки в забезпеченні конфіденційності, а й в забезпеченні контролю справжності та цілісності інформації. Крім того, зростання цінності інформації та інформатизація суспільства ставлять питання розмежування доступу до інформації (наприклад, якщо користувач не сплатив роботу з базою знань) і питання захисту від комп'ютерного тероризму. На сьогоднішній день такий захист здійснюється ефективно з використанням засобів криптографії.

Системи і засоби захисту інформації (СЗІ) відрізняються від "звичайних" систем і засобів тим, що для них не існує простих і однозначних тестів, які дозволяють переконатися в тому, що інформація надійно захищена. Крім того, ефективність СЗІ і просто їх наявність ніяк не пов'язує на працездатності основної системи. Тому завдання ефективності СЗІ не може бути вирішена звичайним тестуванням. Наприклад, для перевірки працездатності системи зв'язку досить провести її випробування. Однак

успішне завершення цих випробувань не дозволяє зробити висновок про те, що вбудована в неї підсистема захисту інформації теж працездатна.

Завдання визначення ефективності СЗІ (особливо, якщо використовуються криптографічні методи захисту), часто більш трудомістка, ніж розробка СЗІ, вимагає наявності спеціальних знань і, як правило, більш високої кваліфікації, ніж задача розробки. Часто аналіз нового шифру є новою науковою, а не інженерною задачею.

Ці обставини призводять до того, що на ринку з'являється безліч засобів криптографічного захисту інформації, про які ніхто не може сказати нічого певного. При цьому розробники тримають криптоалгоритм (як показує практика, часто нестійкий) в секреті. Однак завдання точного визначення даного криптоалгоритму не може бути гарантовано складної хоча б тому, що він відомий розробникам. Крім того, якщо порушник знайшов спосіб подолання захисту, то не в його інтересах про це заявляти. В результаті, користувачі таких СЗІ потрапляють в залежність як мінімум від розробника. Тому суспільству має бути вигідно відкрите обговорення безпеки СЗІ масового застосування, а приховування розробниками криптоалгоритму повинно бути неприпустимим.

РОЗДІЛ 2

2.1 Атаки на архітектуру

Криптографічний система не може бути надійніше використаних в ній окремих алгоритмів шифрування. Іншими словами, для того щоб подолати систему захисту, досить зламати будь-який з її компонентів. Використання хороших будівельних матеріалів ще не є гарантією міцності будівлі. Так і криптографічний система, побудована на основі потужних алгоритмів і протоколів, теж може виявитися слабкою.

Багато системи "втрачають гарантію" безпеки, якщо використовуються неправильно. Скажімо, перевірка допустимості значень змінних не виконується, "випадкові" параметри використовуються багаторазово, що абсолютно неприпустимо. Алгоритми шифрування необов'язково забезпечують цілісність даних. Протоколи обміну ключами необов'язково гарантують, що обидві сторони отримають один і той же ключ.

Деякі системи шифрування, які використовують пов'язані ключі, можуть бути зламані, навіть якщо кожен ключ окремо надійний. Щоб забезпечити безпеку, недостатньо просто реалізувати алгоритм і чекати, що все буде працювати. Навіть наявність кваліфікованих інженерів, допомога відомих компаній і наполеглива праця не можуть гарантувати абсолютної надійності. Проломи, виявлені в алгоритмах шифрування систем стільникового зв'язку стандартів CDMA і GSM, а також в протоколі Microsoft Point-to-Point Tunneling Protocol (PPTP), наочно це ілюструють. Наприклад, в досить надійному алгоритмі RC4, на якому побудований протокол PPTP, вдалося виявити режим, який робив захист абсолютно прозорим.

Ще одне слабке місце криптографічних засобів – генератори випадкових чисел. Розробити хороший генератор випадкових чисел непросто, оскільки він часто залежить від особливостей апаратного і програмного забезпечення [1,2]. Сама система шифрування може бути виконана на високому рівні, але якщо генератор випадкових чисел видає легко вгадуємі ключі, то всі бар'єри

долаються без особливих зусиль. У ряді продуктів використовуються генератори випадкових чисел, що виробляють ключі, в яких простежується певна закономірність. У таких випадках про безпеку говорити не доводиться. Цікаво, що застосування одного і того ж генератора в деяких областях забезпечує необхідний ступінь безпеки, а в інших – ні.

Ще одне можливе слабке місце – взаємодія між окремо безпечними протоколами шифрування [3]. Майже для кожного безпечного протоколу, як правило, можна знайти інший, не менш надійний, який зведе нанівець всі переваги першого, якщо вони обидва використовують однакові ключі на одному і тому ж пристрої. Якщо різні стандарти захисту застосовуються в одному середовищі, недостатньо чітка взаємодія між ними часто може привести до дуже небажаних наслідків.

2.2 Атака на конкретну реалізацію

Багато систем підводять через помилки в реалізації. Деякі продукти не гарантують, що, зашифрувавши текст, вони знищать оригінал. В інших для попередження втрати інформації в разі системного збою використовуються тимчасові файли, а доступна оперативна пам'ять розширюється за рахунок віртуальної пам'яті; в цьому випадку на жорсткому диску можуть залишатися окремі фрагменти незашифрованого тексту.

Переповнення буферів, що не стерта до кінця секретна інформація, недостатньо надійна система виявлення і відновлення після помилок – все це приклади проломів в конкретних реалізаціях, через які дуже часто і проникають зловмисники. У найбільш кричущих випадках операційна система навіть залишає ключі на жорсткому диску. В одному з продуктів великої софтверної компанії введення пароля здійснювався через спеціальне вікно. При цьому пароль зберігався в буфері вікна і після його закриття. Проводити подальші дослідження захищеності системи вже не мало б сенсу.

Слабкі сторони інших продуктів не так явно кидалися в очі. Іноді одні й ті ж дані шифрувалися за допомогою двох ключів: перший з них був надійним, а другий підбирається досить легко; але при цьому експерименти з уже підібраним ключем допомагав підібрати і інший. В інших системах застосовується майстер-ключів і ключі "на один сеанс"; причому безпеці головного ключа приділяється недостатня увага, а основні надії покладалися на одноразові ключі. Для створення по-справжньому надійної системи безпеки необхідно повністю виключити можливість аналізу будови ключів, а не обмежуватися лише найочевиднішими запобіжними засобами.

Творці систем електронної комерції часто змушені йти на компроміс заради розширення функціональності. І оскільки розробники вважають за краще жертвувати безпекою, в захисті раз у раз з'являються дірки. Звірка облікових записів, наприклад, може проводитися тільки раз в день, але за кілька годин зломщик здатний нанести воістину колосальні збитки!

Перевантаження процедури ідентифікації може привести до того, що особистість атаки не буде розпізнано. Деякі системи заносять сумнівні ключі в "чорні списки"; отримання доступу до цих списків істотно полегшує завдання зломщика. Багато системи захисту долаються після повторних атак і використання старих повідомлень або їх частин, які збивають систему з пантелику.

Потенційна небезпека закладена в можливості відновлення раніше використовувалися ключів в системах з розщепленням [4]. У хороших криптографічних системах термін життя ключів обмежується максимально коротким проміжком часу. Процедура відновлення дозволяє продовжити життя ключа вже після того, як від нього відмовилися. Використовувані для відновлення ключів бази даних самі по собі є джерелом небезпеки, і їх архітектура повинна бути вивірена з особливою ретельністю. У ряді випадків проломи в них дозволяли хакерам маскуватися під легальних користувачів.

2.3 Атаки на обладнання

Деякі системи (найчастіше комерційного призначення) мають так зване "кільце безпеки", що складається з апаратних засобів підвищеною стійкістю до зломів (смарт-карт, електронних гаманців, електронних ключів і т.д.) [5,6]. Творці подібних систем виходять із припущення, що архітектура системи всередині цього кільця надійно захищена від несанкціонованого доступу. Надійність обладнання – дуже важлива складова комплексних систем безпеки, але не варто повністю довіряти рішенням, що захищає тільки від злодійства і невмілого поводження.

Більшість подібних технологій на практиці не працюють, а інструменти для їх злому безперервно удосконалюються [5,6]. При проектуванні подібних систем дуже важливо не забувати про додаткові механізми захисту, які повинні спрацьовувати, якщо хакерам вдасться подолати перші оборонні редути. Потрібно постаратися максимально ускладнити завдання противника і зробити її рішення невигідним з економічної точки зору. Вартість даних, що захищаються повинна бути значно нижче витрат на руйнування системи безпеки. Цінність електронного проїзного не може йти ні в яке порівняння з вартістю портфеля цінних паперів. Виходячи з цього і слід проектувати засоби захисту.

У 1995 році значно зросла кількість "атак за розкладом": з'ясувалося, що секретні ключі RSA можна відновлювати, вимірюючи тимчасові інтервали між операціями шифрування [7]. Був зареєстрований ряд випадків успішного злому смарт-карт, а також серверів електронної комерції в Internet. Виявилося, що атаки будувалися на основі вимірювання споживаної потужності, аналізу електромагнітного випромінювання та інших побічних джерел інформації. Фахівцям з криптографії вдалося за цими ознаками реконструювати логіку багатьох систем з відкритими ключами, продемонструвавши їх ненадійність.

Велику популярність придбав метод аналізу збоїв, що дозволяє знаходити слабкі місця кріптопроцесорів і відновлювати секретні ключі.

Подібні методи за своїм духом швидше біологічні. Криптографічні системи в цьому випадку розглядаються як складні об'єкти, які реагують на зовнішні подразники. Їх не можна чітко описати за допомогою математичних рівнянь, але наслідки таких атак руйнівні.

2.4 Атаки на моделі довірчих відносин

Багато способів подолання захисних рубежів пов'язані з моделями довірчих відносин всередині системи. Перш за все, слід виявити зв'язку між окремими компонентами системи, усвідомити обмеження і механізм реалізації схеми довірчих відносин. Прості системи (засоби шифрування телефонних переговорів та інформації на жорстких дисках) використовують елементарні довірчі моделі. Комплексні системи (засоби електронної торгівлі або засоби захисту багатокористувацьких пакетів електронної пошти) побудовані на основі складних (і набагато більш надійних) моделей довірчих відносин, що описують взаємозв'язку безлічі елементів.

У програмі електронної пошти може використовуватися супернадійний алгоритм шифрування повідомлень, але якщо ключі не сертифіковані джерелом, що заслуговує на довіру, і сертифікація ця не може бути підтверджена в реальному часі, безпеку системи залишається під питанням. Деякі торговельні системи можуть бути розкриті за згодою продавця з покупцем або в результаті об'єднаних зусиль декількох клієнтів. В інших системах передбачено наявність засобів забезпечення безпеки, але якість цих коштів ніхто ніколи не перевіряв. Якщо модель довірчих відносин не документована, то в процесі розгортання в продукт можна випадково внести будь-які неприпустимі зміни, після чого стрункість системи безпеки буде порушена.

Багато програмні пакети занадто довіряються захищеності апаратних засобів. Передбачається, що комп'ютер абсолютно безпечний. Рано чи пізно в таку програму проникає "троянський кінь", який підбирає паролі, зчитує незашифрований текст або якимось іншим чином втручається в роботу системи захисту. Розробникам систем, що функціонують в комп'ютерних мережах, слід потурбуватися про безпеку мережевих протоколів. Уразливість комп'ютерів, підключених до Internet, багаторазово зростає.

Система шифрування, яка долається "з боку мережі", нікуди не годиться. Не існує програм, безпеку яких вистояла після того, як противнику вдалося застосувати зворотне проектування. Дуже часто система проектується з розрахунку на одну модель довірчих відносин, а в практичній реалізації фігурує зовсім інша. Прийняті в процесі проектування рішення повністю ігноруються після передачі готового продукту користувачам. Система, яка абсолютно безпечна, коли її оператори заслуговують довіри, а доступ до комп'ютерів повністю контролюється, втрачає всі свої переваги, якщо обов'язки операторів виконують низькооплачувані працівники, найняті на короткий термін, а фізичний контроль за комп'ютерами втрачений.

Втім, хороші моделі довірчих відносин продовжують працювати навіть в тому випадку, якщо окремі компоненти підводять.

2.5 Атаки на користувача

Навіть якщо система гарантує надійний захист при правильній експлуатації, користувачі можуть випадково порушити її, особливо якщо система спроектована недостатньо добре [8]. Класичним прикладом є співробітник, який надає свій пароль колегам з тим, щоб вони мали можливість вирішувати нагальні завдання під час його відсутності. Атака з урахуванням "людського фактора" часто виявляється куди більш ефективною, ніж місяці кропіткого аналізу алгоритмів [9].

Користувачі можуть протягом декількох днів не повідомляти про втрату смарт-карти. Вони не приділяють необхідної уваги перевірці електронного підпису. Секретні паролі часом повторно використовуються в несекретних системах. Клієнти навіть не намагаються ліквідувати слабкі місця в системі безпеки. Звичайно, навіть хороші системи не в змозі ліквідувати наслідки причин соціального характеру, але вони можуть звести їх до мінімуму.

Багато продуктів зламуються тому, що їх захист побудована на основі паролів, що генеруються користувачами. Надані самі собі люди не замислюються про те, як вибрати незвичайну послідовність символів. Адже пароль, який неможливо підібрати, не так просто запам'ятати. Якщо в якості секретного ключа застосовується такий пароль, то підібрати його, як правило, вдається набагато простіше і швидше, ніж використовуючи метод грубої сили.

Багато призначених для користувача інтерфейсів ще більше полегшують завдання зловмисника, обмежуючи довжину пароля 8 знаками, перетворюючи вводиться послідовність в символи нижнього регістра і т.д. Навіть паролі-фрази не забезпечують необхідного ступеня безпеки. Зловмиснику набагато легше підібрати фразу з 40 букв, ніж перебирати всі можливі послідовності 64-розрядних випадкових ключів. Іноді захист, в якій застосовуються дуже надійний механізм ключі сеансів, руйнується через використання слабких паролів, призначених для відновлення ключів. Бажання полегшити

відновлення системи після збою фактично відкриває перед атакуючими чорний хід.

2.6 Атака на засоби відновлення після збоїв

Розробники надійних систем не в змозі закрити в паркані безпеки все найдрібніші щілини, але принаймні зяючі дірки вони ліквідують. Відновлення ключа до одного файлу не дозволить зломщику вважати всю інформацію, що знаходиться на жорсткому диску. Виготовлення фальшивих грошей - дуже серйозний злочин, адже володар технології друкування грошей може знищити національну валюту. Хакер, зламує смарт-карту, вивчає секрети даного конкретного пристрою, а не всіх інших смарт-карт, що входять в систему. У багатокористувацьких системах знання секретів однієї людини не повинно відкривати доступу до інформації інших.

Багато системи за умовчанням встановлюються в режим з відключеними засобами безпеки. Якщо система захисту "заїдає", користувач просто відключає її і продовжує займатися своєю справою. Така поведінка робить особливо ефективними атаки типу denial-of-service ("відмова в обслуговуванні"). Якщо онлайнова система авторизації кредитних карт відключена, продавець змушений задовольнятися значно менш надійною паперовою технологією.

Іноді у зломщиків з'являється можливість скористатися зворотною сумісністю різних версій програмного забезпечення. Як правило, в кожному новому варіанті продукту розробники намагаються усунути прогалини, які були в старому. Але вимога зворотної сумісності дозволяє атакуючому застосовувати протокол старої, незахищеної версії.

Деякі системи не мають коштів відновлення. Якщо захист зруйнована, повернути програму в працездатний стан не представляється можливим. Вихід з ладу системи електронної торгівлі, до якої звертаються мільйони клієнтів, загрожує обернутися катастрофою. Тому подібні системи повинні мати засоби організації протидії атакуючим і підтримувати можливість оновлення системи безпеки без зупинки програми.

Добре продумана система сама знає, як краще протистояти атаці і що слід робити для усунення пошкоджень і оперативного відновлення працездатності.

2.7 Атака на засоби шифрування

Іноді слабкі місця можна знайти і безпосередньо в системі шифрування. Деякі продукти створюються на базі не надто вдалих алгоритмів власної розробки. Як правило, розкрити відомі алгоритми шифрування вдається лише у виняткових випадках. Якщо ж розробник робить ставку на власні методи, шанси зломщиків підвищуються багаторазово. Незнання секрету алгоритму не є особливою перешкодою. Кваліфікованого фахівця досить пари днів, щоб по об'єктному коду відновити вихідний алгоритм шифрування.

Надійність стандартної для електронної пошти архітектури S / MIME 2 не в змозі компенсувати слабкостей алгоритму шифрування. І без того не надто надійний захист GSM від слабкого алгоритму шифрування програє ще більше. У багатьох системах використовуються занадто короткі ключі [10].

Можна навести безліч інших прикладів помилок в системах шифрування: програми повторно генерують "унікальні" випадкові значення, алгоритми цифрового підпису не в змозі забезпечити контроль за переданими параметрами, хеш-функції відкривають те, що повинні захищати. У протоколи шифрування вносяться не передбачені розробниками зміни. Користувачі люблять "оптимізувати" наявні кошти, доводячи їх до такого примітивного рівня, що вся система захисту руйнується, як картковий будиночок.

ВИСНОВКИ

Аналіз використовуваних на практиці криптоаналітичних атак для дешифрування комп'ютерних програм показує, що зашифровані вихідні коди програм, а також виконувані коди програм є особливо уразливими по відношенню до атак з використанням відомого відкритого тексту і підібраного відкритого тексту. Кріптоаналітку досить зробити дані типи атаки на основі припущення, що в початковому тексті програми використовуються відомі ключові слова і стандартні ідентифікатори, відомі рядкові повідомлення, а виконувані коди програм містять відомі коди команд, виклики функцій і багато-багато іншого.

Радикальним способом атаки на криптосистему є метод повного перебору ключів шифрування. Час, необхідний для повного перебору, залежить від двох параметрів: числа тестованих ключів і тривалості кожного тесту. Тимчасова складність методу повного перебору дорівнює O (2n), де n - довжина ключа.

Слід зауважити, що ще кілька років тому потужність комп'ютерів робила можливими заяви авторів систем захисту про те, що повний перебір ключів шифрування неможливий через величезну кількість можливих ключів. Ще й зараз для демонстрації ефективності системи захисту призводять потужність безлічі ключів (паролів), а надійність підтверджують довжиною ключа. Але сучасні потужності комп'ютерів і новітні технології обчислень дозволяють здійснювати повний перебір навіть досить довгих ключів протягом прийнятного проміжку часу.

Крім того, для збільшення швидкості перебору вже запропоновані і можуть бути вдосконалені ефективні алгоритми пошуку і порівняння (як правило, засновані на формальній логіці і використовують теорію множин, теорію ймовірностей і інші області математики). Завдання повного перебору вирішується за допомогою паралельних процесорів. Кожен процесор тестує особливе підмножина простору ключів. При цьому істотно те, що немає

необхідності в обміні повідомленнями між процесорами, досить одного єдиного повідомлення про успіх; не потрібно і спільний доступ до пам'яті. В особливих випадках можливе використання спеціалізованого устаткування, що виконує функції перебору.

Різновидом методу повного перебору ключів є метод, за допомогою якого розкривають осмислений пароль, так звана атака за словником. Програми, які здійснюють атаку по словнику, досить швидко працюють, так як реалізовують ефективні алгоритми пошуку і порівняння. Багато з них навіть не містять базу слів, а користуються словниками, вбудованими в поширені текстові редактори.

Для збільшення надійності криптографічного захисту в даний час розробляються нові методики, засновані на фундаментальних законах фізики. Так, в даний час розвиваються квантові системи шифрування. Квантова система генерує код, створений з наборів окремих фотонів з різною поляризацією і іншими властивостями. Напрямок, в якому відбуваються коливання електричного поля фотона, відповідає нулях і одиницям. Квантове шифрування також має свої недоліки, тому в даний час багато вчених ставлять під сумнів факт забезпечення високого рівня захисту інформації за допомогою цього методу (особливо при передачі на великі відстані).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1 P.Gutmann, "Software Generation of Random Numbers for Cryptographic Purposes", Proc. 1998 Usenic Security Symp., Usenix Assoc., Berkeley, Calif., 1998, pp. 243-257.

2 J.Kelsey, B.Schneier, and D.Wagner, "Protocol Interactions and the Chosen Protocol Attack", Security Protocols, 5th Int'l Workshop, Springer-Verlag, New York, 1996, pp. 91-104.

3 C.Hall et al., "Side-Channel Cryptanalysis of Product Ciphers", Proc. ESORISC 98, Springer-Verlag, New York, 1998.

4 H.Abelson et al., "The Riscs of Key Recovery, Key Escrow and Trusted Third-Party Encryption", World Wide Web J., No. 3, 1997, pp. 241-257.

5 R.Anderson and M.Kuhn, "Tamper Resistance: A Cautionary Note", Proc. Second Usenix Workshop Electronic Commerce, Usenix Assoc., Berkeley, Calif., 1996, pp. 1-11.

6 J.McCormac, European Scrambling Systems, Baylin Publications, Boulder, Colo., 1997.

7 P.Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DDS and Other Systems", Proc. Crypto 96, Springer-Verlag, New York, 1996, pp. 104-113.

8 R.Anderson, "Why Cryptosystems Fail", Comm. ACM, Nov. 1994, pp. 32-40.

9 I.Winkler Corporate Espionage, Prima Publishing, Placer County, Calif., 1997.

10 M.Blaze et al., "Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security", Oct. 1996