hakin9

EXTRA

Issue 6/2012 (13) ISSN 1733-7186

TIMING ATTACKS IN AES

SIDE CHANNEL ATTACKS

CACHE TIMING ATTACKS

AUTOMATED ALGEBRAIC CRYPTANALYSIS

TIMING ATTACK AGAINST THE CBC OPERATING MODE

PLUS

TIMING ATTACKS AGAINST RSA ENCRYPTION AND DECRYPTION REVISITED

Atola Insight

That's all you need for data recovery.

Atola Technology offers *Atola Insight* – the only data recovery device that covers the entire data recovery process: *in-depth* **HDD diagnostics**, **firmware recovery**, **HDD duplication**, and **file recovery**. It is like a whole data recovery Lab in one Tool.

This product is the best choice for seasoned professionals as well as start-up data recovery companies.

Emphasized features at a glance:

- Automatic in-depth diagnostic of all hard drive components
- Automatic firmware recovery and ATA password removal
- Very fast imaging of damaged drives
- Imaging by heads
- Case management
- Real time current monitor
- Firmware area backup system
- Serial port and power control
- Write protection switch

Atola TECHNOLOGY

Visit atola.com for details

Managing:
Michał Wiśniewski
m.wisniewski@software.com.pl

Senior Consultant/Publisher:
Paweł Marciniak

Editor in Chief:
Grzegorz Tabaka
grzegorz.tabaka@hakin9.org

Art Director:
Marcin Ziółkowski

DTP:
Marcin Ziółkowski
www.gdstudio.pl

Production Director:
Andrzej Kuca
andrzej.kuca@hakin9.org

Marketing Director:
Grzegorz Tabaka
grzegorz.tabaka@hakin9.org

Proofreadres:
Dan Dieterle, Michael Munt,
Michał Wiśniewski

Top Betatesters:
Ruggero Rissone,
David von Vistauxx,
Dan Dieterle,
Johnette Moody,
Nick Baronian,
Dan Walsh,
Sanjay Bhalerao,
Jonathan Ringler,
Arnoud Tijssen,
Patrik Gange

Publisher: Hakin9 Media Sp. z o.o. SK
02-682 Warszawa, ul. Bokserska 1
www.hakin9.org/en

Dear Readers,

This month we decided to prepare a spacious issue on Timing Attacks. There are two reasons for that: first – as an "extra" branch of hakin9 we search for the hottest topics in IT-Security and we enjoy expanding on the topics that we have prepared. The second reason, however, has everything to do with the launch of the newly established Cryptomag. We are preparing completely new magazine, independent of hakin9, and solely devoted to cryptography (as its name suggests). Stay tuned to hakin9 news and be ready for the new magazine when it appears. Below is what we have prepared for you in this month's hakin9 Extra. Vincent Rijmen in his article on "Timing Attacks on AES" will show you how the execution time of an AES encryption can be used to derive the secret key. Qi Chai, in this issue's special article, will re-visit Timing Attacks against RSA. Weizhong Yang and Jeffrey Zheng are going to present Variant Pseudo-Random Number Generator. Michael W. Farb, Yue-Hsun Lin, Adrian Perrig and Jonathan McCune are going to expatiate on Safeslinger – an easy-to-use and secure public-key exchange. Martin Rublik, or regular collaborator is going to present an Overview of Side Channel and Timing Attacks. In an article entitled "The Dichotomy of Symmetric vs Asymmetric Cryptography" Wayne Patterson discusses the fundamental dilemma of the two kinds of cryptography in today's use. Matthieu Bontrond is going to present Timing Attack against CBC operating mode – an attack that enables decryption of blocks without attacking the encryption key. Theodosis Mourouzis has presented us Automated Algebraic Cryptanalysis. Michael Wisher presented his expertise on "Cache-Timing Attacks on Symmetric Cryptographic Primitives". Nitin Jain is going to present you the article on "Timing Attacks on Practical Quantum Cryptographic Systems. The last, but not least is the interview with Vitaliy Mokosiy – Atola's Bandura Project Manager and key developer.

I hope that you will enjoy the reading!

Michał Wiśniewski, hakin9 Extra
m.wisniewski@software.com.pl

MONITOR STRONY

Innowacyjne e-usługi do monitorowania stron www

SEOmonitor
monitorowanie strony www na potrzeby SEO

SPEEDmonitor
monitorowanie prędkości ładowania strony www

CONTENTmonitor
monitorowanie poprawności językowe treści publikowanych na stronie www

www.monitorstrony.pl

INNOWACYJNA GOSPODARKA
NARODOWA STRATEGIA SPÓJNOŚCI

DOTACJE NA INNOWACJE
Projekt współfinansowany przez Unię Europejską
z Europejskiego Funduszu Rozwoju Regionalnego

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ
ROZWOJU REGIONALNEGO

TIMING ATTACKS ON AES

VINCENT RIJMEN

Abstract: The black-box security of AES remains unchallenged. Nobody is able to decrypt ciphertexts that have been encrypted with AES and neither is there a method known to recover the secret key, even when a large amount of messages and the corresponding ciphertexts are known. However, if the attacker has access to additional information about the internal operations of AES, then practical attacks are sometimes possible. In this article, we explain how the execution time of an AES encryption can be used to derive the secret key.

Introduction

It has been known since a long time that programs can pass on hidden information deliberately by varying their consumption of CPU resources, e.g. execution time or RAM usage. However, only in 1996 Paul Kocher published the first article on *timing attacks*: methods that recover the secret key of otherwise practically unbreakable cryptographic algorithms by exploiting detailed information on their execution time [4].

In this article, we explain two timing attacks on AES. Firstly, by way of introduction, we show how a naive implementation of the finite field operations used in the MixColumns step of AES leads to a simple attack. This attack can also be avoided easily. Next, we show an attack based on the timing differences caused by the working of cache memory. The attack assumes that an attacker can make accurate timing measurements and requires a bit more analysis, but is also more difficult to counter.

The execution time of a program is just one type of *side-channel information*. In particular for implementations in hardware and on very simple processors (think: smartcards), researchers discovered powerful attacks based on measurements of the power consumption or the electro-magnetic radiation during an encryption operation. Since those attacks require a thorough understanding of the design of electronic circuits, we won't cover them here.

We assume that the reader is familiar with the AES (Advanced Encryption Standard). Otherwise, good descriptions can be found in the FIPS standard [1] and in the AES book [2].

Simple case: xtime

An implementation of MixColumns

The MixColumns step of the AES round transformation contains multiplication operations in the finite field GF(256). Each byte of the input is multiplied by the constants **1**, $\boldsymbol{x}$ and $\boldsymbol{x} \oplus \mathbf{1}$. (The last two constants are often denoted by **2** and **3**.) On a typical processor, these multiplications are implemented by means of a table lookup. In this example, however, we assume that the implementation explicitly computes the multiplications. This could be the case, for instance, if the processor has so little RAM (or cache) available, that we don't want to store this table.

The multiplication by $\boldsymbol{x}$ can be implemented as shown in Algorithm 1. Note that multiplication by $\boldsymbol{x} \oplus \mathbf{1}$ can be implemented by using the law of distributivity in GF(256): $a \times (\boldsymbol{x} \oplus \mathbf{1}) = (a \times \boldsymbol{x}) \oplus (a \times \mathbf{1}) = (a \times \boldsymbol{x}) \oplus a$. Hence, an implementation of xtime and some xor operations are all that we need to implement MixColumns.

Visual inspection of Algorithm 1 reveals that on a simple processor without advanced scheduling tricks, there will be a noticeable variation in the execution time of this routine. If the MSB of a is set, then the routine will take longer, because that branch contains extra instructions.

A timing attack consists now essentially of three phases:

- Trigger AES encryptions of different message blocks. Measure and record the execution times.

TIMING ATTACKS AGAINST RSA REVISITED

QI CHAI

The prevailing belief -- an information system is secure due to the employment of cryptographic functions that are mathematically strong -- can go wrong if adversaries does not play by the presumed rules. In fact, attacks may happen in completely unexpected ways such as compromising the cryptographic functions through measurements of the time they take to accomplish certain tasks, known as the *timing attack*. In this article, we exhibit instructive cases and examples of how to attack a weak class of implementations of the most popular public-key cryptographic algorithm, i.e., RSA, by making wise use of the runtime it reveals during operating. The lesson learnt is that the theoretic security of crypto algorithms should be examined in conjunction with the implementation security of them that may add another layer of complexity to the development of secure systems.

1 Introduction

1.1 What is RSA?

Before 1976, symmetric-key encryption, where the secret keys used by the encrypter, i.e., Alice, and the decrypter, i.e., Bob, are identical (or can be simply transformed from the one to the other), is the only known paradigm to protect data. To share the encrypted information with other parties, the secret key has to be distributed or delegated, which could be problematic, e.g., how to deliver the key effortlessly and secretly without introducing additional encryptions? how could one ensure that each of the key-holders will keep the key information privately from unauthorized parties especially in the long run?

This problem has been thoroughly solved thanks to Diffie and Hellman's breakthrough invention of the public-key cryptography, which, based upon the intractability of some computational hard problems, could accomplish the tasks like encryption and decryption using two different keys -- one is published and the other is private. Anyone

inj3ct0r

if you'll hacked us
we'll pay you 10K $
http://1337day.com/

Exploit database separated by exploit type
(local, remote, DoS, Poc, etc.)

VARIANT PSEUDO-RANDOM NUMBER GENERATOR

WEIZHONG YANG, JEFFREY ZHI J. ZHENG

Abstract – Variant Pseudo-Random Number Generator (VPRNG) based on the Variant Logic framework – an extension of Cellular Automata (CA) - is proposed to construct a PRNG. A list of classical methods on PRNG, BBS, ANSI X9.17 and DES were used in comparison under the NIST Statistical Test Suite, the measurement results show that the VPRNG can produce a better pseudo-random number series in most cases than the compared models.

Introduction

The security of cryptographic systems depends on secret data that is known for authorized persons but unknown and unpredictable to others [1,3,5]. To meet this unpredictability, some randomness mechanisms are required [4,8]. Quality in Pseudo Random Number Generation PRNG is required for security in both stream cipher and block cipher applications [3,4,7,8], and lack of quality generally provides attack vulnerabilities [3,8]. From a practical viewpoint, some key properties for good randomness mechanisms are essentially important such as period length, efficiency and ease of implementation [1,3,7].

For some PRNGs, the period length can be calculated without walking through the whole period. For a PRNG internal state contains *n* bits, Linear Feedback Shift Registers LFSRs usually have periods of exactly $2^n - 1$ Normally systems based on essential Boolean operations are in higher performance and better efficiency in hardware or firmware implementations than compared systems based on arithmetical operations [3,8,9].

The PRNG system is particularly attractive to attackers because it is typically a single isolated component easy to be located in environment. Different cryptanalytic attack technologies are applied to PRNG mechanisms such as guessing of seed, timing attacks on state advance function, output generation functions, forward and backward tracking attacks [2,5-7].

To secure wider applications, it is a challenge task to design and implement a proper PRNG to have a list of superior properties [1-9].

In this paper, a new PRNG system – Variant Pseudo-Random Number Generator VPRNG – based on [12] is proposed. Combining different technology [10-12] with additional extensions, this system can provide extreme longer circular properties for a *n* bit state, VPRNG have a basic period of $2^n \times 2^{2n}$ bits for a configuration and also a huge variation space of $2^n \times 2^{2n}$ configurations available to use efficient tabular operations with significantly superior behaviors in comparison to other PRNG results. The system of VPRNG is described in section II-VI.

Variant Pseudo Random Number Generator

The architecture

The architecture of Variant Pseudo-Random Number Generator VPRNG is shown in Figure 1.

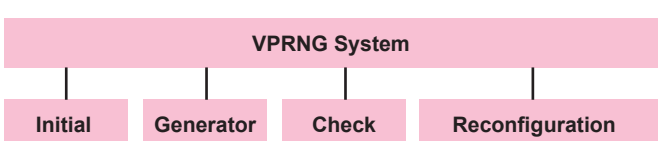


Figure 1. *The Architecture of Variant PRNG*

In the architecture, there are four modules: Initial, Generator, Check and Reconfiguration. Further detailed descriptions on various parameters are discussed in Sections III-VI respectively.

In the Initial module, it provides a list of initial arguments for the Generator, such as the Cell Serial X as a seed, the rule R, the complement Δ and the permutation P operators respectively.

Using the rule R, the Generate module selects a complementary rule Δ and a permutation rule P and works on Cell sequence to generate new Cell series. Under a given configuration, this module can provide a non-repeat sequence with a length of $2^n \times 2^{2n}$ bits [12].

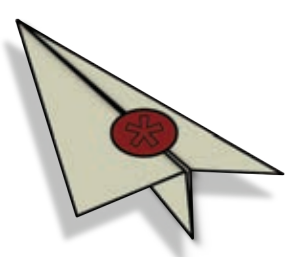
In the check module, it focuses on checking the current Cell series in which whether its content has or not to be repeated. If it does not repeat, then it will be passed.

SAFESLINGER:
EASY-TO-USE AND SECURE PUBLIC-KEY EXCHANGE

MICHAEL W. FARB, YUE-HSUN LIN, ADRIAN PERRIG, JONATHAN MCCUNE

For many current Internet applications, users experience a crisis of confidence. Is the email or message we received from the claimed individual or was it sent by an impostor? Cryptography alone cannot address this problem. We have many useful protocols such as SSL or PGP for entities that already share authentic key material, but the root of the problem still remains: how do we obtain the authentic public key from the intended resource or individual? The global certification process for SSL is not without drawbacks and weaknesses, and the usability challenges of decentralized mechanisms such as PGP are well-known.

Of course, ordinary users can extensively rely on system administrators' help in making trust decisions. However, ordinary users inevitably face challenging decisions alone; most users at home, on travel, on vacation, or in small businesses do not benefit from skilled help. All this while the need and temptation to use new online services steadily increases.



The recent proliferation of smartphones offers a promising opportunity to address these challenges, as these devices offer a general computing environment with a powerful processor, high-resolution display, several communication modalities (Bluetooth, WiFi, 3G/4G), camera, and sensors.

Unfortunately, smartphone platforms suffer from many risks. Vulnerabilities exist in communication standards that enable eavesdropping or impersonation. Moreover, phone operators are disclosing information or introduce vulnerabilities through insecure or misconfigured systems.

Individuals often have physical interactions with resources or other individuals before communicating digitally. Often, people communicate over the Internet or via SMS after having met in person. We leverage this physical encounter to bootstrap digital trust.

Solution Roadmap

SafeSlinger is a system for secure exchange of authentic information between two smartphones, and a user interface for secure messaging. In essence, SafeSlinger exchanges contact information, containing public keys in addition to standard contact list information such as name, picture, phone numbers, email addresses, etc. Thanks to the association between the individual holding the phone and the public key that is exchanged, users (with the help of the SafeSlinger App) can later associate digital communication with the previously met individual by verifying a digital signature. To make SafeSlinger usable, the cryptographic aspects are mostly hidden from the user, and we have built-in several approaches to make SafeSlinger tolerant to user error.

We envision SafeSlinger as a general approach to bootstrap secure digital communication. (1) First, we've enabled groups (2-9 individuals) of physically co-located users to securely bootstrap trust by sending keys (**slinging keys**) between their devices (a one-time operation). SafeSlinger can also support remote setup, as long as users can authenticate the other individual (e.g., via telephone communication or live video conference). (2) Second, SafeSlinger supports secure phone-to-phone messaging and file transfer, providing both secrecy and authenticity. Once users' devices hold each others' public keys, the SafeSlinger user experience is nearly identical to that of traditional SMS and MMS messaging today. (3) Third, we will enable secure introductions without physical meetings by allowing a common acquaintance to facilitate a mutual introduction enabled by SafeSlinger's file transfer. (4) Fourth, we plan to release our source to enable other applications to adopt the SafeSlinger API to add their public key to a contact entry. Now, when a user sends (**slings**) its updated contact

list entry to another user, each application's public key is automatically included, and the same application at the other end can extract the public key. This mechanism can enable applications such as secure email or secure SMS to solve the problem of securely exchanging the public key without a leap of faith.

We have implemented SafeSlinger on Android and iOS, and it can be installed from their respective app stores to enable secure credential exchange. The public apps include mechanisms for secure messaging and file exchange for Android, with plans to release the same functionality for iOS in Summer 2012. Beyond that, we will extend SafeSlinger to provide secure introductions between two individuals.

Attacks to Resist

The main purpose of SafeSlinger is to enable a set of users to exchange their contact information such that every non-malicious user receives the correct information about every other non-malicious user. Malicious users may collude and impersonate each other, for example, therefore we cannot provide any guarantees for those parties. Our main goal is to provide high usability while preventing the attacks described later.

Secure local exchange of information is a surprisingly intricate and challenging problem. Possible attacks include:

- *Malicious bystander who participates in protocol:* a bystander can overhear conversation, and attack the protocol by controlling the local wireless communication. (Dolev-Yao attacker model). The Man-in-the-Middle (MitM) attack is a specific instance of this attack.
- *Malicious group member:* an invited member of the group wants to violate protocol properties, such as mounting an impersonation attack by injecting incorrect information for another user, or performing a Sybil attack by injecting multiple entries either for fictitious individuals or for individuals who are not present. A malicious group member can also perform a Group-in-the-Middle (GitM) attack, as described above.
- *Malicious server:* for protocols that rely on a back-end server, the server may be controlled by a malicious administrator or become compromised.
- *Information leakage after protocol abort:* an adversary may be able to cause a protocol abort and trigger leakage of private information about a participant.
- *Collision attack on low-entropy hash:* as described above, low-entropy hash values can be vulnerable to efficient birthday attacks if precautions are not taken.

Challenges to Overcome

We have also considered the following challenges, some of which directly apply to SafeSlinger, and others which we sidestep via the design choices that we make for SafeSlinger:

- *Exclusion of unintended participants:* legitimate users will expel an unwanted bystander who wants to participate in the protocol.
- *Correct member count:* users need to correctly count the number of group members who participate in an exchange.
- *Identity validation:* users correctly validate the identity information received from the exchange. They should map the identity information to the people who participate in the exchange, and they should reject information of non-participants.
- *Impersonation detection:* users verify that no other user has injected information that impersonates them in the current exchange. For example, a malicious user may also inject information about Alice, even though Alice is also participating in the exchange. The risk is that another user may discard the correct information and accept the wrong information.
- *Diligent hash comparison:* users can correctly perform a hash comparison, even after executing the protocol numerous times without any attack.
- *Diligent error checking and aborting:* users will abort the protocol and restart the protocol when suspicious or error conditions are encountered.

User Experience

Although the protocol is complex, the user experience is actually quite simple as SafeSlinger performs all the cryptographic operations and checks without the users' involvement.

The user experiences the following steps: (1) select the data items to be shared, (2) count and select the number of users, (3) find and enter the lowest ID displayed by the devices, (4) compare the word phrases and select the one that matches and click "match", or click on "no match", (5) select which users' data to import into its contact list.

Protocol Steps

In a nutshell, the mobile devices send their information to the server, which redistributes it to the other devices. The users then engage in a verification of all exchanged information to ensure that they all have received identical information. This verification is finished by the users who perform a comparison of textual representation derived from short hash values displayed by the phones.

Two problems must be avoided here: (1) users may habituate to click "OK" without performing the comparison, and (2) an attacker may compute a collision attack on the short hash value.

We solve (1) by presenting 2 decoy hash values alongside the correct value and asking users to verify which of the 3 hash values matches a value on other people's devices. We address problem (2) by using Short Authentication Strings (SAS), where, all devices first commit to the values that are used in the hash comparison. Once all the commitments are distributed, the devices reveal the decommitments and the short hash comparison can proceed. This approach prevents the collision attack and in Zimmermann's words converts the attack from a "safe attack" into a "daring attack."

Another challenge that we address in SafeSlinger is to prevent the server from learning any contact information. We accomplish this by leveraging a *group Diffie-Hellman protocol*. The group DH protocol establishes a shared secret key among all participants, which is used to encrypt the contact information. To prevent MitM attacks, the DH public key is included in the initial commitment and thus validated during the hash comparison.

In the following we go through the SafeSlinger protocol within each steps:

- *Data Selection & Counting:* the user selects which data to share and enters the total number of protocol participants.
- *Commitment, Group DH Key Setup:* each device computes the values needed for the group DH protocol by selecting the DH private key n_i at random. The device also randomly selects nonces to indicate "match" (Nonce match Nm_i) and

"wrong" (Nonce wrong Nw_i). The device also encrypts the data D_i to share with Nm_i used as a symmetric encryption key (using AES with 128-bit keys): $E_i = \{D_i\}_{Nmi}$. We also use SHA-1 truncated to 128 bits as hash function H. Figure 1 depicts this multi-value commitment structure for user U_i. Finally, the device sends the commitment C_i to the server.

- *Server Unique ID Assignment, User Grouping:* In the next step, the server groups the users. First, the server sends a unique ID to the device which the device displays and prompts the user to find the **lowest ID** amongst all devices. Then all users enter the lowest ID to their device and send it to the server.
- *Collection and Distribution of Initial Decommitment:* The server now knows which devices belong to the same group, and distributes ID and commitment pairs (ID_i, C_i) to all group members. Once a device receives all commitments, it opens up the first level decommitment (HN_i, G_i, E_i) (See Figure 1). If validation of all decommitments is correct, devices compute a hash over all decommitments of C_i, i.e., over the triplets (HN_*, G_*, E_*), sorted by the value of the unique ID_i assigned to each device to ensure that all devices compute the hash over the same triplet ordering.
- *Wordlist-based Comparison of Integrity of Commitments:* Each device then computes a word phrase that represents the hash -- we use the PGP word phrase which results in a representation that encodes 24 bits of the hash. If no phrase matches, the user selects "no match" and sends the "wrong" nonce Nw_i along with Hm'_i to enable verification to the server. This case is also triggered if the user selects the wrong word phrase. This approach cryptographically authenticates the "no match" message from the commitment C_i and thus prevents injection of the wrong nonce

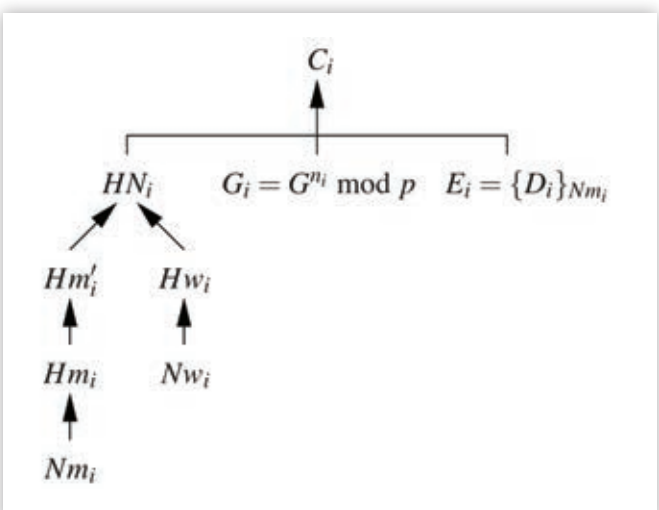


Figure 1. *Multi-value commitment structure for User U_i. C_i, HN_i, Hm'_i, Hw_i, Hm_i, Nm_i, are 160-bit values; G_i is 512 bits, and Di varies in length. Each arrow imples SHA-1 hash operations.*

by an adversary. Later, users correctly selected the matching word phrase, and the device reveals the pair of values indicating success (Hm_i, Hw_i), which the server redistributes to other devices.

- *Group DH Key Establishment:* Each device can verify that all the users selected the correct word phrase and the devices proceed to construct the group DH tree. The ordering in the tree is determined by the sorted order of the unique ID_s ID_*. Since the group DH protocol we use is intricate, we omit the details for enhanced readability.
- *Distribution and Verification of Data Decryption Key:* Once the secret group key K is established, the devices then proceed to send their final match nonce Nm_i (which serves as the data decryption key) to the group, encrypted with K.
- *Decryption of Data and User Acquisition:* In the end, each device decrypts and verifies the correctness of Nm_i, and finally uses Nm_i to decrypt the data D_i.

Architectural Features

SafeSlinger provides several convenient and useful features based on its design.

- *Fast, Consistent, Wireless Communication:* We currently target Android and iPhone devices, with an effort to make design decisions compatible with future implementations for other platforms (e.g., Windows Phones). Unfortunately, today's platforms do not offer consistent support for 802.11 ad-hoc mode or seamless creation of a base station to enable other devices to connect to them. Bluetooth communication is also inconvenient because of the slow discovery phase and the inability of iPhones to communicate with non-Apple devices (excepting headsets). NFC is not yet widely deployed, and such communication does not scale beyond pairwise communication. As a consequence, we use Internet-based communication, where all the mobile devices connect to a cloud server. This approach has the additional advantage that no latency for device discovery is experienced, as the devices can simply send packets to the server via IP.
- *Scale:* Though the current technical limit for our protocols and implementation is much greater than 9 users, it is unclear that there is much value in scaling even this far due to human limitations. We concentrate our presentation on groups of up to 9 users, leaving it as an open question whether there is a need for protocols that scale further. As SafeSlinger protocol fails to complete if only a single person miscounts, we set the threshold at 9 users. Asking users to count the number of participants rules out several attacks.
- *Grouping:* When mobile devices initially connect to the server, the server does not know which devices belong to the same group. It is a challenging problem for the server to determine the grouping, especially if several concurrent exchanges are ongoing. We employ the following approach that does not leak any sensitive information to the untrusted cloud server. The server assigns a unique ID to each mobile device, which it displays to its user. The devices prompt the users to find and enter the lowest ID. The devices then send that ID back to the server, which can thus perform the grouping. Note that the actual grouping is not security-sensitive, as an intruder can only cause denial of service.
- *Confidentiality During Data Exchange:* All exchanged data is encrypted and the actual encryption key K_D and initialization vector IV_D are derived from the 160-bit "match" nonce Nm_i. Contact data integrity is achieved through verification of the commitment C_i, hence, no additional Message Authentication Code (MAC) is needed. Since we can validate Nm_i based on the commitment Hm_i, no additional MAC value is needed to ensure integrity and authenticity for that encryption.
- *Word Phrase Verification:* In SafeSlinger, the word phrase is constructed from the first 24 bits of the 160 bit SHA-1

hash. We use the standard PGP approach for converting a 24-bit value into 3 words. PGP uses two word lists -- an "even" and "odd" list -- with 256 words each. Based on the standard PGP approach, the first 8 bits select a word in the "even" list, the second 8 bits select a word in the "odd" list, and the final 8 bits select another word from the "even" list. We discourage careless comparison by displaying two unique (across all devices in the exchange) decoy phrases in addition to the common phrase.

- *Word Phrase Collision Avoidance:* Although unlikely, the words in a decoy phrase may match the words in a decoy phrase on another device, causing the user to select the decoy phrase which results in an error detected by the local device. We want to avoid true randomness in the decoy phrases so that careless users will not chose the wrong phrase if the actual hash phrase and either of the decoy phrases contain the same word in the same position. We thus chose our decoy phrases deterministically such that each decoy word will be unique across all decoy phrases displayed in the group.
- *Address Book Key Management:* We rely heavily on the mobile operating systems' contact list facilities to manage users' contact data and public keys. It is convenient to store users' public key data in a recognizable field in the smartphone's address book, so that any existing synchronization service will seamlessly maintain backups. We realize this functionality by adding the name and value (base-64 encoded public key) of a new instant messaging (IM) provider to the contact list.

Applied Key Exchange

We have implemented SafeSlinger as a base API library on both Android and Apple iOS platforms. In the future, any third-party application for either platform may link against or execute the key exchange (with its GUI).

Cryptographic operations are computed using the operating system-provided libraries for Android and iOS, with the addition of open source OpenSSL libraries for Apple iOS. Figure 2 shows the information flow between multiple devices during execution of the key exchange, outlined as follows.

- 3rd-party app generates a public/private key pair.
- 3rd-party app inserts its public key in the device's contact list.
- 3rd-party app invokes the key exchange API, specifying the appropriate contact list entries (with the name of the public key(s) to exchange).
- During the SafeSlinger key exchange protocol, multiple messages are exchanged between devices via our server, and validated independently by each device.
- The newly received (and authenticated) public key(s) and contact data are saved in the device's contact list.
- 3rd-party applications may now make use of newly imported public keys from the contact list.

Applications

We have implemented secure rich messaging for Android and iOS. Secure information and shared public keys via Key Exchange are used to encrypt and authenticate text messages and file data. When SafeSlinger has been installed, it generates an RSA 2048-bit key pair first. The application then obtains a Google Android C2DM Push token or an Urban Airship token for addressing the Android or iOS devices.

During a SafeSlinger information exchange, the push tokens of all group members are exchanged and imported into the address book alongside the public keys.

Some interesting potential uses for SafeSlinger are:

- Secure Text Messages: Separate corporations which already manage internal employees, each with a large PKI, who want to share sensitive materials, do not have to choose one internal PKI to use, as SafeSlinger Messaging can bridge the gap between large but separate existing PKIs.
- Secure File Transfer: Hospitals that want to remotely share test results or imaging data with their doctors or patients, can do so confidentially through a SafeSlinger secure file exchange on the patient's phone.

Secure Introductions

A future implementation could leverage our secure file transfer mechanism to enable secure introductions. A common friend of two users sends contact data that includes public keys, to each other. More concretely, consider Alice with two friends: Bob and Carol. Alice has performed a SafeSlinger exchange with both Bob on one occasion, and with Carol on another, and has thus received an authentic SafeSlinger public keys for both Bob and Carol. Likewise, both Bob and Carol have Alice's authentic SafeSlinger public key. In a secure introduction, Alice first encodes Bob's contact information (which includes Bob's SafeSlinger public key and Push token) as a custom vCard and uses an OpenPGP message format to provide secrecy and authenticity. Alice then sends this message via SafeSlinger to Bob. Hence, Bob can validate that the information indeed originates from Alice, whom he trusts not to send bogus information. Analo-

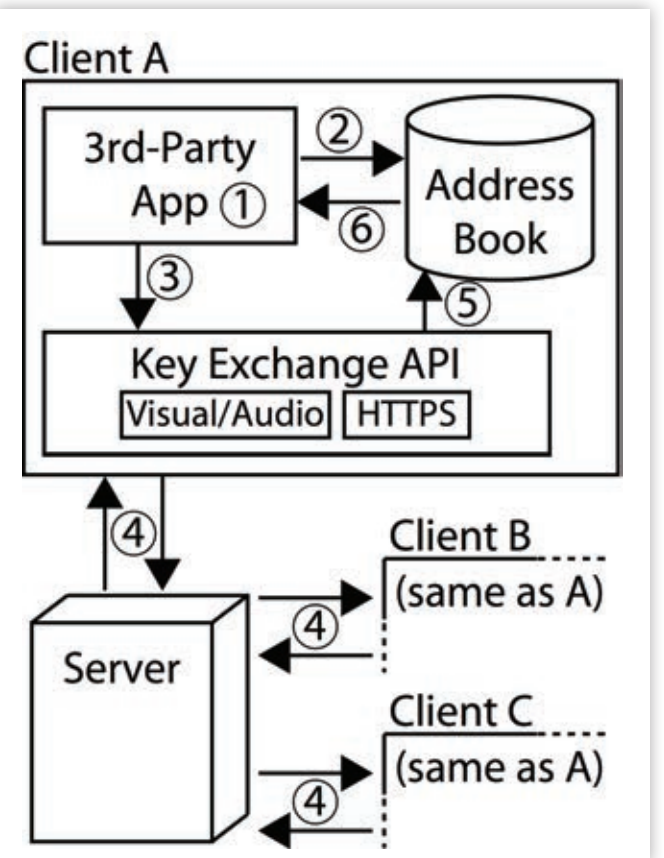


Figure 2. *SafeSlinger keyexchange API interactions with 3rd-party applications.*

gously, Carols trusts Bob's information received from Alice. Now that Bob and Carol have each other's public keys and push tokens, they can use SafeSlinger to securely communicate.

Summary

To realize the vision of secure online communication, we need to overcome several human challenges: some users are ambivalent about security or privacy, most users lack security expertise, and many users prefer convenience over security and may not want to expend much effort for security. To counteract these challenges, we designed SafeSlinger as an easy-to-use application that offers many benefits to drive usage.

We have released our SafeSlinger application both for Android and iOS devices with the intention to provide a free and easy-to-use system that enables secure communication. Through free multi-platform applications available on smartphone markets, open documentation, and open-source code, we anticipate wide adoption of SafeSlinger. Assuming wide adoption, we hope to provide usable and secure communication for the masses, and a security platform that will enable numerous security services and applications. A more detailed technical white paper[1] can be found at our website[2], and our applications can be installed from the Google Play Store and iTunes App Store.

MICHAEL W. FARB

joined Carnegie Mellon CyLab as a Research Programmer in 2010. He received his BA from Beloit College in 1995, and as a mobile device software developer, has worked in publishing, transportation, and security.

YUE-HSUN LIN

joined Carnegie Mellon Cylab as a Postdoctoral Researcher in 2012. He received his PhD from National Tsing Hua University in 2010. His research includes wireless security, sensor network security, and secure protocols design.

ADRIAN PERRIG

is a Professor in Electrical and Computer Engineering, Engineering and Public Policy, and Computer Science at Carnegie Mellon University. Adrian serves as the technical director for Carnegie Mellon's CyLab. He earned his Ph.D. degree in Computer Science from Carnegie Mellon University.

JONATHAN MCCUNE

is a Research Systems Scientist for CyLab at Carnegie Mellon University. He earned his Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University, and received the A.G. Jordan thesis award. He received his B.Sc. degree in Computer Engineering from the University of Virginia (UVA).

[1] *http://sparrow.ece.cmu.edu/group/pub/SafeSlinger.pdf*
[2] *http://www.cylab.cmu.edu/safeslinger*

OVERVIEW OF SIDE CHANNEL AND TIMING ATTACKS

MARTIN RUBLIK

Attacking a system is a tradeoff between the attacker's possibilities and gains and be-tween the difficulty of an attack. The article will focus on special techniques that attacker can use to gain valuable information only by observing the system. Though these types of attack are not easy to execute, the attacker can gain a lot. These attacks can be used to subvert a system that is secure under a commonly accepted threat model. A system can be secure in theory, but the at-tacker can subvert the system's defense by exploiting the vulnerabilities that were either not included in the threat model or emerged during the implementation phase.

Introduction

Based on what the attacker knows about the target we can divide the attacks into several categories:

- attacks where the attacker does not know anything about the target,
- attacks where the design of the target system is known,
- attacks where implementation of the target system is known or
- attacks where details about the implementation and environment specific details are known (such as software and hardware equipment, services topology, etc.).

Some systems rely on obscurity as a defense mechanism and this is where the first class of attacks belongs. In general this is not a good tactic, especially in cryptography. Modern cryptographic systems are designed with a premise that the internals of the system are known to the attacker. The reason is that it is really hard to keep secret about the design of a system, as attacker can usually reverse engineer the system easily. Several examples support this design principle. Among the others the most known failure was design of DVD CSS algorithm used for DVD protection. This algorithm was broken by Jon Lech Johansen who was only 16 year old at the time.

Attacking the system design is mostly a theoretic task, but breaking it, has severe consequences to the system and its practical use. In cryptography these types of attack are mostly algorithmic attacks and are of course implementation independent. Therefore when a *practical* algorithmic attack on cryptographic system is found the system needs to be replaced where applicable. An example of such an attack would be design flaws in WEP [1] that lead to WPA/WPA2 rollout or flaws found in MD5 hash algorithm [2] that lead to global hash algorithm change in X.509 certificates.

The third class of attacks is implementation specific and in general can be "easily" patched. However the patching possibilities may vary depending on the system's characteristics. For example it is easier to patch a network connected server operating system vulnerability, but it is much harder to patch an offline embedded system.

An example of implementation specific flaw would be the infamous Debian cryptographic random number generator flaw that allowed the attacker to brute-force a private cryptographic key generated on a Debian server [3] easily. The issue was that the key-space used for generating the private keys was reduced less than 300K keys due the bug in the random number generator. PRNG was incorrectly seeded by process IDs ranging from 0 to 32767 providing not enough seed entropy (only 15 bit) for generating the private keys.

Finally attacks, where special hardware or software equipment is known to the attacker are applicable only in special situations. On the other hand attacker has most knowledge about the targeted system, thus he can exploit system in several ways. An example of implementation and environment dependent attack on a system is a side channel attack.

Side channel attacks

Side channel attack is an attack where attacker infers valuable information by observing the system's behavior. In general side channel attack analyses leaked information with regards to hardware, software, implementation and design specifics of the targeted system.

Side channel attacks can be either passive or active. In a passive side channel attack the targeted system leaks information within the ordinary communication and the attacker deduces the results only by observing the system. In an active side channel attack the attacker can actively manipulate the

THE DICHOTOMY OF SYMMETRIC VS ASYMMETRIC CRYPTOGRAPHY

WAYNE PATTERSON

Two roads diverged in a yellow wood,
And sorry I could not travel both …

Robert Frost, "The Road Not Taken", 1920.

In this article we will describe a fundamental dilemma in the world of cryptography, because of significant differences in the two types of cryptography in use today. The concept of symmetric cryptography is at least as old as Julius Caesar, and has been the only approach throughout history until the last generation. Asymmetric cryptography was only conceived in the 1970s, but it solves certain problems that cannot be addressed in the symmetric mode. Yet asymmetric cryptography introduces difficulties of its own. The challenge of resolving these two approaches at the algorithmic level is wide open.

Cryptology is an ancient field of study, with its origins going at least back to the era before Christ. However, whereas encryption for centuries dealt with transformations of symbols from natural languages, the invention of the computer introduced a more important symbol set as the basis for cryptology, the set {0, 1}.

Furthermore, with the use of binary symbols and the underlying environment for transforming messages becoming the computer and the networks to which computers were attached, new problems arose that eventually realized the famous quote from Robert Frost cited above. And so cryptology has faced its diverging paths, and one might say that the road has not yet been chosen, and may never do so.

In order to describe this dichotomy between two differing approaches that we will define as *symmetric* and *asymmetric* cryptography, it will be necessary to review the techniques developed over centuries and why in certain instances they have failed to provide solutions in the context of modern-day communications.

The first two millennia of symmetric cryptography

Historically, most encryption systems have been based either on the concept of substitution, or of transposition, or both.

Substitution

One of the earliest known cryptosystems is usually referred to as the *Caesar shift,* after Julius Caesar [1]. The technique Caesar shift uses is a simple substitution of the symbols used in communication. For simplification, let us suppose that we are encrypting a message (called the *plaintext*) that uses the 26 symbols from our Roman alphabet. In order to encrypt, we will write the letters in order, and then advance them by a fixed number of positions in the alphabet, understanding that the letter following Z will be A, and so on. Thus the letters of the alphabet in the encryption will be substituted by the corresponding letter shifted the chosen number of positions, leading to an encrypted message or *ciphertext*. For example, if the number of positions shifted is eight, the correspondence and hence the substitution will be the following:

TIMING ATTACK AGAINST THE CBC OPERATING MODE

MATTHIEU BONTROND

Early 2000, a really nice job has been performed by the EPFL Security and Cryptography Laboratory to study the security of the TLS/SSL protocol. In 2001 and 2002, Serge Vaudenay started to warn the scientific community on security flaws induced by the CBC padding in various security protocols (ref 1 & 2). In 2003, a practical attack has been implemented in a joint work with some of his students and collaborators (ref 3). The CRYPTO'03 presentation provides a clear explanation of the process and description of the limits.

This attack enables decryption of blocks without attacking the encryption key. Compared to classical cryptanalysis techniques, the amount of computation and trial and error attempts are very reasonable. These properties may render this attack very efficient against SSL-like protocols; however today's protocols' configuration have a great impact on this attack.

This attack exploits properties of different components which are often implemented in communication protocols:

- A characteristic of the CBC operating mode and weak padding types,
- An inherent property of the "MAC then Encrypt" philosophy,
- Error messages produced at a decryption error event.

"MAC then Encrypt" philosophy

The "MAC then Encrypt" philosophy consist in computing some integrity check value on data to be sent before the ciphering process. This is a widely adopted process but the reverse process "Encrypt then MAC" is more relevant in term of security because information leakage will occur potentially less easily.

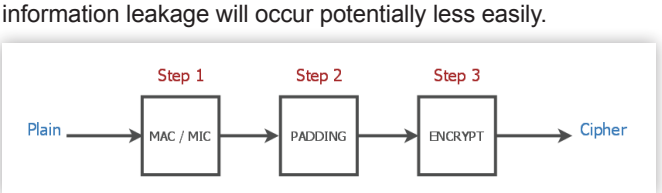


Figure 1. *"MAC then Encrypt" encryption process*

The reverse process takes place in the reverse order, thus the validity of the padding scheme is checked before data integrity.

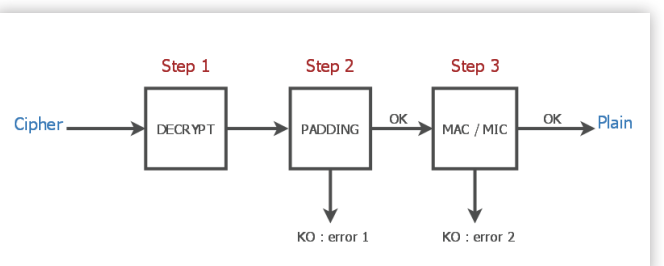


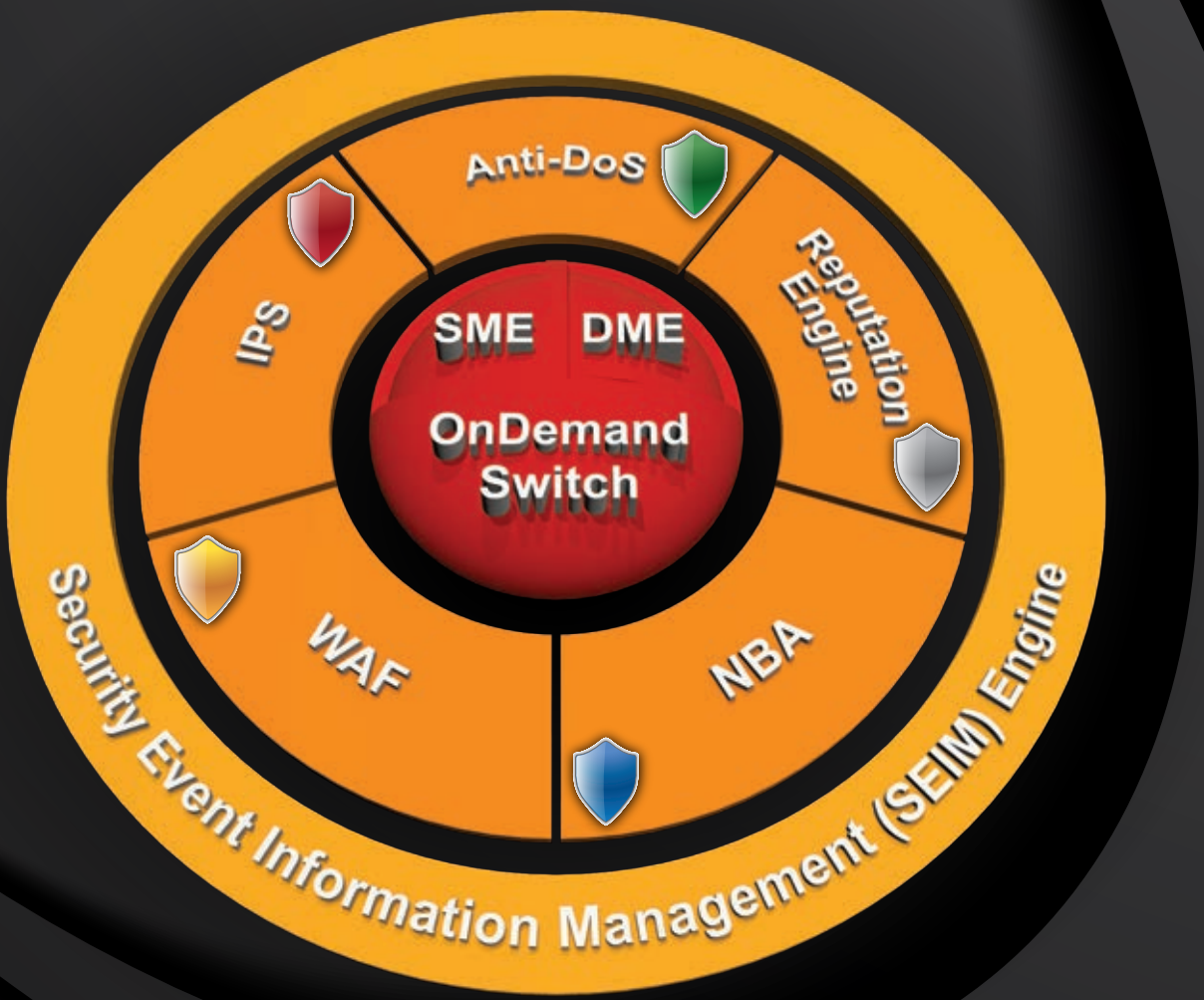
Figure 2. *common "MAC then Encrypt" decryption process*

As a consequence, potential error messages may be produced at different stages. In particular integrity check require more resources to be computed than the padding scheme to be controlled. This holds even more if the data integrity is ensured by a strong cryptographic function. This remark is an entry point to perform a timing attack. The attack exposed by the researchers consists in exploiting the possibility to differentiate padding errors and integrity check failures.

Error messages

When this problem has been discussed between people from OpenSSL team and security researchers of EPFL to implement

radware



Protecting Networks from a New Age of Hacktivism

Radware Attack Mitigation System:

- Real-time, Multi-vector Attack Detection
- Hardware Accelerated DDoS Mitigation
- Integrated Event Correlation & Reporting
- 24x7 Emergency Response Team Support

For more information, visit: www.radware.com

AUTOMATED ALGEBRAIC CRYPTANALYSIS

THEODOSIS MOUROUZIS

Abstract – Algebraic attack is a form of known-plaintext attack and consists of two basic steps. Firstly, one converts the given cryptographic primitive into a multivariate system of polynomial equations usually over or any other algebraic system and then tries to solve for the secret key. The first step is called *modelling* while the second step *solving*.

Crypto-designers' aim is that the underlying system of equations is not solvable faster than exhaustive key search. In general, solving a random multivariate system of equations is *NP-hard* [11]. However, in most cryptographic schemes, their rich algebraic and geometric properties can be further exploited to solve the underlying system.

In this article, we provide an introduction to algebraic cryptanalysis and we describe how this 2-step process can be considered as an automated cryptanalytic process. Such attacks have been a big success for stream ciphers, however for block ciphers, until recently, only a limited number of rounds could be broken. In the last section we present a key recovery algebraic attack for 4 rounds of the Russian government standard block cipher GOST [7] given 2 known pairs of plaintexts and ciphertexts [13].

Keywords – Cryptanalysis, algebraic attack, NP-hard, Multivariate System, Algebraic Normal Form (ANF), Conjunctive Normal Form (CNF), Multivariate Quadratic (MQ), SAT, GOST block cipher

Introduction

Cryptanalysis of block-ciphers is divided into two main classes; the **structural attacks** and the inversion or **generic attacks**. Structural attacks exploit the particularities of a cipher due to its design and the specific properties of its underlying components. Generic attacks are form of black-box attacks and are general purpose algorithms that solve multivariate systems of equations. If we manage to solve this very complex system of equations and obtain the secret key , then we launched a successful algebraic attack against the system. Algebraic attacks apply to a variety of ciphers, ranging from blockciphers, like AES and Serpent [2], to stream-ciphers, like Toyocrypt [9] and Bluetooth [8], and asymmetric cryptosystems, like HFE [10].

Algebraic Cryptanalysis is a subfield of cryptanalysis, whose success relies on the fact that some block ciphers exhibit a high degree of algebraic and geometric structure. It is a *known-plaintext* attack and consists of the following two steps:

Step 1: (MODELLING)
Express the cipher operations as a multivariate system of polynomial equations over or any other algebraic system in terms of key plaintext and ciphertext bits.

$$\{f_1(K,P,C)=0, f_2(K,P,C)=0,..,f_r(K,P,C)=0\} \Leftrightarrow E(K,P)=C$$

,where are functions describing cipher's operations.

Then substitute all known pairs in order to decrease the complexity of the system by eliminating some variables, resulting in equations involving only bits of the secret key For example if we are given one known pair and we can form equations in the key bits , then given another pair we obtain equations in the key bits , increasing the probability that the system can be solved. We assume that the encryption is executed under the same key for the given pairs.

Step 2: (SOLVING)
Solve the underlying multivariate system of polynomials and obtain the secret key.

The idea of algebraic cryptanalysis is not new. Shannon in his paper "*Communication theory of secrecy systems*", states that "***Breaking a good cipher should require as much as solving a system of simultaneous equations in a large number of unknowns of a complex type***",[1].

In general, solving a random multivariate system of equations is known to be NP-hard and thus it is not surprising that all ciphers can be expressed into a system of polynomial equations [11]. That does not imply at all that these systems are solvable faster than exhaustive key search. However, not all the multivariate systems are NP-hard and especially in the area of cryptography some algebraic or geometric properties of the ciphers can be further exploited in order to solve the underlying system of equations.

Design of Systems: Cryptosystems which are designed based on the computational hardness of solving a random multivariate system of equations in finite fields are called ***Mul-***

CACHE-TIMING ATTACKS ON SYMMETRIC CRYPTOGRAPHIC PRIMITIVES

MICHAEL WISHER

Many attacks on cryptographic algorithms target flaws in the algorithm designs. The flaws can be exploited by intercepting ciphertext, and measuring whether it has statistical biases. From the biases, the attacker assigns probabilities to the different potential keys that generate the ciphertext. Although common in academic literature, this type of attack rarely is practical, since it requires very large amounts of ciphertext generated under a single key. Side-channel attacks consider both the design and implementation of an algorithm. Side-channel leakage is additional information that allows the complexity of an attack to be reduced significantly. Cache timing attacks, t a-c

Side-channel attacks versus theoretical attacks

According to the traditional principles of cryptography, many experts would consider the Advanced Encryption Standard (AES) block cipher, or its Chinese equivalent, SMS-4 to be broken if someone found a statistical attack that could recover a 128-bit key using less than 2^{128} encryptions. Although theoretically broken, the running time of an attack with complexity 2^{120} encryptions is infeasible and the attack could never be completed.

But with access to the cryptographic device, and the ability to precisely inject a single fault into the block cipher chip using a laser, more than 100 bits of the SMS-4 block cipher master key can be easily recovered using just a handful of ciphertexts [1]. The remaining bits of the key can be efficiently guessed.

This style of attack, differential fault analysis, is an active side-channel attack, in which the attacker manipulates the state of the cryptographic device in order to derive the side channel information. The assumptions for this style of attack are very strong – it is possible that someone who stands in front of the device with the ability to manipulate it so precisely is able to read the key directly from its memory. Unless the fault is only transient, the attack is probably detectable.

It is much more difficult to detect passive side-channel attacks, in which the attacker does not affect the device, but uses additional information – such as noise, timing, electromagnetic signals – in addition to the stream of intercepted ciphertext.

Because of the power of this class of attacks, side-channel attacks have recently become a hot topic. Many types of side-channel attack require considerable technical knowledge of the implementation platform, and the details of the attack will vary according to the platform. The topic of this article, timing attacks, mostly allows potential vulnerabilities to be detected during the design process, and algorithm designers can do much to alleviate the vulnerability of their algorithms at the design stage.

Timing attacks

Timing attacks are passive attacks. The side-channel for timing attacks is the difference in the amount of time that it takes to execute different operations or blocks of operations. On some machines, the speed with which a primitive operation can be completed might differ according to the value of the operand.

TIMING ATTACKS ON PRACTICAL QUANTUM CRYPTOGRAPHIC SYSTEMS

NITIN JAIN

With photons being the only available candidates for long-distance quantum communication, most quantum cryptographic devices are physically realized as optical systems that operate a security protocol based on the laws of quantum mechanics. But to finally yield a stream of bits (secret key) usable for encryption, a quantum-to-classical transition is required. Synchronization of electronic & optoelectronic components involved in such tasks thus becomes a necessary and important step. However, it also opens up the possibility of timing-based loopholes and attacks.

Introduction

In a letter to Max Born written in 1926 [Born, 1969], Albert Einstein remarked: "Quantum mechanics is certainly imposing. But an inner voice tells me that this is not yet the real thing. The theory says a lot, but does not bring us any closer to the secrets of the Old One. I, at any rate, am convinced that He is not playing dice." This quote, particularly the last part about God not playing dice, indicates Einstein's unwillingness to accept a fundamental tenet of quantum theory: with regards to values of physical quantities, only statistical assertions can be permitted. Indeed, Einstein and some other prominent scientists were also inclined towards the more classical view of the world in which physical systems could be ascribed properties that existed irrespective of whether they were being measured or not [EPR, 1935]. It was thus believed by some that quantum mechanics could not provide a complete description of Nature.

Fast forward to the next century, and with principles of quantum mechanics having been verified in innumerable different experiments, it seems that the earlier view of those scientists was incorrect. Nonetheless, due to its bizarre nature and ideas, quantum mechanics still confounds anyone who tries to understand it. But thankfully, that hasn't stopped us from

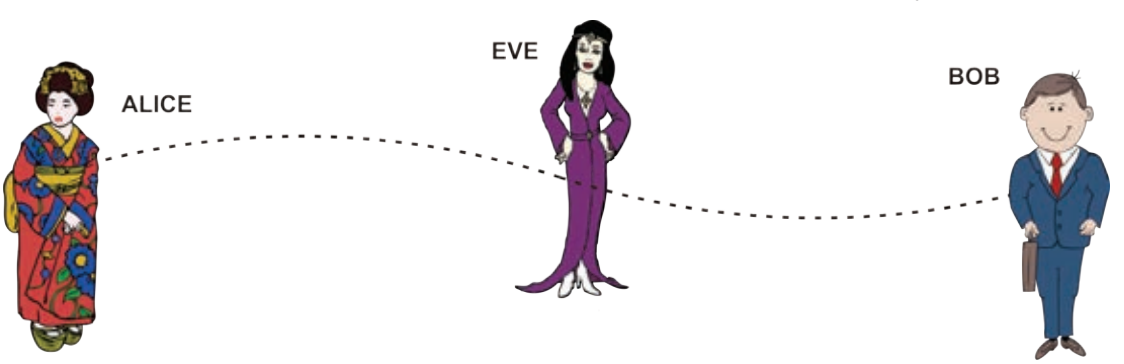


Figure 1. *Fundamental scenario for cryptography:* Two entities, normally called Alice and Bob wish to share a secret which a third party, usually called Eve (who usually also harbours some malicious intent) is also interested in knowing.

STAFFCOP

PC monitoring, Corporate Security and Data Loss Prevention Software

StaffCop Standard allows you to monitor all activities on company computers and prevent the unauthorized distribution of sensitive corporate information.

StaffCop will help you:

To locate possible data loss channels and prevent loss
To gain insight into how your employees spend their work time
To increase company and departmentals efficiency

You need StaffCop to:

Gather work time efficiency statistics
Easily control your employees in real-time mode
Improve discipline and motivation of your employees

Who needs StaffCop:

CEO/CTO
Corporate Security Manager
HR Manager
System Administrator

Main Features of StaffCop:

Screenshot recording
Application monitoring
E-mail monitoring
Web site monitoring
Chats/IM activity recording
USB device monitoring
Clipboard monitoring
Social Networks Monitoring
Search Term Tracking
File and Folder tracking
Keystroke recording
System Event Monitoring
Whitelists and Blacklists
PC activities reporting
Stealth installation/monitoring
Strong security
Alert notifications
Remote Install / Uninstall

More Information, Demo Versions, Videos and Technical Guides -

www.STAFFCOP.com

Phone: +1-707 -7098405
Skype: staffcop.com
Email: sales@staffcop.com, paul@atompark.com

Microsoft CERTIFIED Partner

Do You Want to Become a Cyber Security Expert?

OR ADVANCE YOUR IT SECURITY CAREER?

- Cyber Security has one of the largest market shares in IT
- Government & Compliance Regulations are more and more enforced
- Gartner Group predicts unprecedented growth and need in Cyber Security
- Skilled Cyber Security Experts are in ever more demand

THE CYBER 51 EXPERT COACHING FORUM

- Individual 1-on-1 Mentoring on Ethical Hacking, Penetration Testing and IT Security
- Networking with other community members and moderators
- Access to a wealth of tools and information not found on public domain
- Permanent Job & Contract offers, Webinars and much more!

YOUR BENEFITS

- Become an Ethical Hacker / Penetration Tester with 1-on-1 mentoring
- Learn at your own pace at a fraction of the cost of regular courses

CYBER 51 COACHING FORUM

CYBER SECURITY FORUM

CONTENT:

1. General Topics
2. Service Assessment
3. Ethical Hacking
4. Cyber Threats
5. Mitigating Cyber Threats
6. Penetration Testing

CYBER 51 INSTRUCTORS

OUR CERTIFICATION LEVELS:

- Certified Ethical Hacker (C|EH)
- Forensic Investigator (C|HFI)
- Certified Security Analyst (ECSA)
- Licensed Penetration Tester (C|LPT)
- Network Security Admin (ENSA)
- ISC Consortium (CISSP)

FEATURES

ADDITIONAL FEATURES:

- 1-on-1 Coaching
- Trainers with Years of Experience
- Wealth of Tools
- Webinars
- Networking with other members
- Contract & Perm. Job Opportunities

WHY CYBER 51?

- Learn whenever you want to
- Dedicated 1-on-1 Coaching
- Information you will not find on public boards
- All Mentors work as Senior Security Consultants
- Frequent updates
- Great Value for money

CYBER 51

A PURE CYBER SECURITY NETWORK

CONTACT US TODAY

CYBER 51 LIMITED, 176 THE FAIRWAY, SOUTH RUISLIP, HA4 0SH, MIDDLESEX, UNITED KINGDOM

EMAIL: INFO@CYBER51.CO.UK

WEB: WWW.CYBER51.CO.UK

ATOLA BANDURA: SUPERFAST IMAGER, WIPER, AND TESTER

An interview with Vitaliy Mokosiy, the Atola Bandura project manager and head developer

VITALIY MOKOSIY

is the Atola Bandura project manager and architect. He has been working in Atola Technology as an expert in software development of HDD tools for data recovery and forensics since 2008. Vitaliy is also known as the project manager of Atola Disk Recycler and as the lead software developer of Atola Insight. His success in all projects is a mix of more than nine years of .NET and Java development experience and team management capabilities.
LinkedIn profile: http://www.linkedin.com/in/vitaliymokosiy

Vitaliy Mokosiy is the lead Atola Bandura developer. He is an expert in the development of HDD tools for data recovery and forensics. Vitaliy kindly agreed to share some information about Atola Bandura with dearly beloved readers: its history, development, opportunities, and advantages in discovering the tool's value.

What is the history of Atola Bandura? What made you decide on its development? Who are the main developers?
As for me, the development process of Atola Bandura itself was extremely exciting. During the process, I encountered lots of interesting things, starting from idea conceptualization to our first market delivery.

The great success of Atola Insight project lead by Dmitry Postrigan has brought us to designing a new and innovative system. We expected Bandura to possess the following features:

- Full-color, easy to use touch screen user interface (UI)
- High-quality duplication of damaged hard disks without connection to a PC (stand-alone mode)
- Maximum possible imaging and wiping speed

Thus, we strived to make the process of testing, duplicating, or erasing of any HDD very easy and straight-forward for our users. As a result, Atola Bandura was specifically designed as a high-speed and easy-to-use imager. It's like a Swiss army knife that includes many additional features (disk diagnostics, checksum calculation, disk comparison, bad sector repair, HPA/DCO, etc.).

How much time did you spend on designing? Did you have any specific difficulties?
In general, the period from idea conceptualization to delivery took about eighteen months. Our team that developed the first edition of Atola Bandura consisted of two hardware engineers, two software developers, and two quality assurance engineers.

Reimagined for Small Business

COMODO ENDPOINT SECURITY MANAGER 2.0

Centrally manages the deployment of award-winning Comodo Internet Security software to protect the desktops and servers your business relies on against internal and external threats.

COMODO®

www.comodo.com

Develop for the Next Big Platform!

Attend the Windows Phone Developer Conference and get the best developer training!

WP DevCon

The Windows Phone Developer Conference

October 22-24, 2012
Hyatt Regency
Burlingame, CA
www.WPDevCon.net

Learn from the top experts at the Windows Phone Developer Conference, including 12 Microsoft MVPs!


Darrin Bishop


Michael Cummings


Nick Landry


Jose Luis Latorre


Chris Love


Colin Melia


Walt Ritscher


Lino Tadros


Kelly White


Shawn Wildermuth


Chris Williams


Chris Woodruff

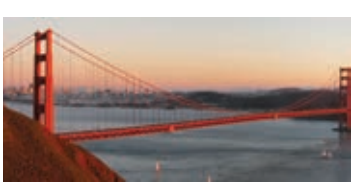
50+ Classes and Workshops

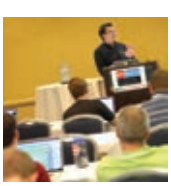
focus on a variety of important topics:

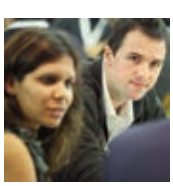
- Design implementation
- Location intelligence services
- Rich data visualization and implementation
- Cloud-based mobile solutions
- Development leveraging HTML5
- User experience
- Application design
- HTTP protocol
- Building reusable components
- Microsoft push notification service
- Creating custom animation
- and many more!

Visit WPDevCon.net for a full list of speakers, bios, classes, workshops, and special events!

Learn, network, and seize the opportunities that Windows Phone represents.







Register Early for the biggest discounts! at www.WPDevCon.net

Produced by BZ Media SD Times @WPDevCon