Saurabh Shintre

An Information-theoretic Approach to Side-channel Analysis

MAP-tele Doctoral Programme em Telecommunications

Departamento de Engenharia Eletrotécnica e de Computadores
Faculdade de Engenharia da Universidade do Porto

December 2015

Saurabh Shintre

An Information-theoretic Approach to Side-channel Analysis

Tese submetida à Faculdade de Engenharia da Universidade do Porto para obtenção do grau de Doutor em Telecomunicações

Orientadores:
Professor Doutor João Barros, Universidade do Porto, Portugal
Professor Doutor Virgil Gligor, Carnegie Mellon University, Pittsburgh, USA

December 2015

Acknowledgements

The financial support for the research presented in this thesis was provided by Fundação para a Ciência e a Tecnologia (FCT) Portugal (Grant no: SFRH/BD/71350 /2010), CyLab, and the John and Clair Bertucci Foundation. I would like to thank my committee members, Prof. Radha Poovendran (University of Washington, Seattle), Prof. Rohit Negi (Carnegie Mellon University) and Prof. Aníbal Ferreira (University of Porto, Portugal) for agreeing to be part of the thesis committee. The feedback and comments provided by them led to significant improvement in the thesis document.

The Ph.D. has been a long and arduous journey with numerous ups and downs. The end has only been possible due to the support and encouragement of advisers, family members, and loved ones. I would like to thank my advisers, Prof. Virgil Gligor and Prof. João Barros, for taking a chance on me and fighting a lot of battles on my behalf. I appreciate the patience and faith they showed in me during difficult periods of the Ph.D.. Their support went well beyond their academic and administrative roles and over the years, they have also guided me during difficult personal times. I hope to enjoy the privilege of their counsel in the future as well.

My parents, Shashikant and Anjali, have my eternal gratitude for everything they have done for me. They inculcated the sense of curiosity in me, and encouraged me to be fearless in the pursuit of my ambitions. All this while, they expected nothing in return and have largely experienced my successes from far. While the Ph.D. did not allow me to spend much quality time with them, I strive to makeup for it in future. My elder brother, Vaibhav, has been my problem-solver since childhood and I would like to express my gratitude to him and my sister-in-law, Madhura, for their support. Sharva, my two-years old niece, has my thanks for unknowingly providing me with joy through her laughter and silly poems.

My best friend and wife, Shiksha, is someone I owe a lot to. She believed in me to finish the Ph.D. even when I did not. She was my support when I could not turn to anyone else for help. I would like to thank her for all the wonderful travels, strange gifts, and helping me explore a a new side of my personality. Time spent with her will always have a special place in my heart. I would also like to express my gratitude to her parents, Johns and Sunita Mantri, sister, Khushboo, and brother-in-law, Nishant, for making me a part of their family and for their unconditional love.

Friends have made this journey immensely pleasurable. In these distant worlds, they became an extended family and I will forever cherish the moments spent eating, drinking, hiking, climbing, and playing cricket, soccer, squash, etc. with them. My special thanks to my different house-mates, Vishnu, Raghav, Hari, Aaditya, Luis, Antonio, and Carlos for putting up with my occasional craziness and listening to me ramble about things for hours. Other special friends who deserve a special mention for helping me out during problems, big and small, include Vinay, Sahil, Prajna, Dhruv, Uday, Ashwati, Amit, Vikram, Divya, Yash, Tanmay, Balaji, Charan, Prithvi, Vignesh, Madhu, Damião, Max, and Niranjini. Being a sport enthusiast, a lot

of my time was spent in such activities, which allowed me to form life-long friendships. For all those great memories, I would like to thank my cricket team-mates, Sameen, Kashif, Pritish, Anoop, Vishal, Ravi, Yaseen, Ayaz, Randip, Shiva, Sahil, and Agneev. My climbing-mates, Supreeth, Utsav, and Varun have my special thanks for teaching me an activity that I thought I could never learn. For interesting philosophical discussions and parties, I would like to express my gratitude to quiz-club members , Keshav, Chitra, Vinod, Varuns, Utsav, Harini, Srini, Sidharth, and Ranjini. My lab-mates Minsuk, Miao, Jun, Zongwei, Chang-han, Adwait, Soo-jin, Max, Rajat, Chad, Zachary, Janos, and others have my thanks for their crucial technical and professional help. Back in Portugal, the combined role of colleagues and friends was donned by Mate, Sanja, Pedro, Rui, Damião, Mari, multiple Joãos, Hana, Tiago, Susana, and Luis. I thank them for helping me settle into a new country, ordering my food, dealing with paperwork, and making Porto a home away from home for me.

An equally difficult aspect of this Ph.D. was to deal with bureaucracy over two countries, two universities, and multiple funding agencies. The system could have broken down at any time without the help of the wonderful people in the CMU Portugal program, CyLab, Instituto de Telecomunicação Portugal, Carnegie Mellon University and the University of Porto. My personal thanks to Sara Brandão, Prof. João Claro, Prof. José Moura, Lori Spears, Samantha Goldstein, Silvia Bettencourt, Carlos Ferreira, Alexandra Vieira, Toni Fox, Brittany Frost, Ivan Liang, Nathan Snizaski, Karen Lindenfelser, and others for dealing with tons of paperwork to keep me in the program.

Abstract

Side-channels are unanticipated information flows that present a significant threat to security of systems. Quantitative analyses are required to measure the rate of information leakage and the accuracy of information learned through side-channel attacks. To this end, the work presented in this thesis develops a general model of a side channel, which is represented as a two-input-single-output system and specified by the probability distribution of the output conditioned on the inputs. For this model, three quantitative metrics are defined: *capacity*, *leakage*, and *reliability rate*. The thesis argues that *capacity* is an ill-suited metric for side channels and recommends the use of other two metrics to measure the leakage rate and accuracy of information learned, respectively. These metrics are used to analyze attacks employed in very different application areas: private communication detection in VoIP networks, packet schedulers in web communication, and timing attacks against modular multiplication routines used in public-key cryptosystems. The analyses presented in this thesis enable us to: 1) determine system parameters and user behaviors that preserve privacy, 2) compute the lifetime of private information, and 3) identify attack strategies that leak most information. More importantly, they enable us to study the conditions under which existing countermeasures perform as expected and develop information-theoretic countermeasures against side-channel attacks.

Resumo

Canais colaterais são fluxos de informao produzidos por um sistema, imprevistos aquando da sua especificação ou implementação, que constituem ameaas significativas á sua segurana. A taxa de fuga e exactidão da informação extraída através de um ataque a um canal colateral são medidas através de análises quantitativas. Esta tese apresenta um modelo geral de um canal colateral, representado por um sistema de duas entradas e saída única e especificado pela distribuição da probabilidade condicional da saída dadas as entradas. Este modelo compreende três métricas quantitativas: capacidade, fuga e taxa de fiabilidade. Esta tese defende que a capacidade é uma métrica inadequada para canais colaterais e recomenda a utilização das outras duas métricas para medir a taxa de fuga e exactidão da informação extraáda, respectivamente. Estas métricas são usadas para analisar ataques aplicados em diferentes contextos: a detecção de comunicação privada em redes de VoIP; programadores de pacotes em comunicação web; e ataques de temporização contra rotinas de multiplicação modular, usadas em sistemas de criptografia de chave pública.

A análise apresentada nesta tese permite: (1) determinar parâmetros do sistema e comportamentos dos utilizadores que preservam privacidade; (2) calcular o tempo de vida de informação privada; e (3) identificar estratégias de ataque que resultam altas taxas de fuga de informação. Acima de tudo, a mesma análise permite estudar em que condições as contramedidas existentes atingem o desempenho esperado; assim como desenvolver contramedidas contra ataques de canal colateral, baseadas em Teoria de Informação.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A *side channel* is an unintended information flow created when the internal functioning of a system is correlated to a physical attribute, such as timing, power consumption, and electromagnetic/acoustic radiation. An attacker who observes any of these attributes can deduce secret parameters, such as cryptographic keys and private system states. The class of attacks that employ side channels is known as *side-channel attacks* (SCAs). To discover a secret parameter of a system, a side-channel attacker issues inputs to the system and observes corresponding change in the attribute. Since the presence of a side channel is typically unknown to the designers and hence often left un-countered, it has the potential of leaking private information even in systems where traditional attacks, like cryptanalysis, fail.

One of the earliest recorded side-channel attacks was performed during the Suez crisis in 1956. In a project code-named ENGULF, the British intelligence agency, MI5, bugged the Egyptian embassy in London using microphones and recorded clicking sounds of the rotors of their mechanical ciphers [66]. Using these recordings, MI5 was able to learn positions of rotor 2 and 3 of the cipher, greatly reducing the complexity of breaking it. They used similar techniques to break the cipher of the French embassy by observing the electromagnetic leakage of its tele-printer. Declassified documents from the U.S. National Security Agency (NSA) revealed a similar program named TEMPEST that used emanations from electro-mechanical devices to break cryptosystems [20]. While successful, these attacks still required the attacker to have physical access to the cipher, which limited their practicability. Hence, SCAs received little mainstream attention in security research historically.

SCAs gained prominence after Paul Kocher used timing attacks to break several modern asymmetric cryptosystems [37]. In his seminal paper, Kocher showed that asymmetric cryptographic algorithms consume different amounts of time depending on the plaintext and secret key. He used this vulnerability to break systems, such as RSA [54] and Diffie-Hellman [15], which have been immune to traditional cryptanalysis. The practical effectiveness of timing attacks was strongly established when their feasibility was demonstrated remotely against popular cryptographic libraries, like OpenSSL [6], and devices, such as smart-cards [13].

More recently, the scope of SCAs has increased greatly beyond cryptographic algorithms. Traffic-analysis attacks that use side-channel observations like packet length and timing characteristics, are potent tools in breaching communication anonymity and privacy. These attacks have been successfully demonstrated to extract private health and medical information of web-application users [10]. SCAs can be remotely launched to breach communication anonymity of internet users [28], even when anonymous networks like Tor [47] are employed. Side channels

present in communication end-devices have been exploited to reveal call-record information of user's of private networks [35] [34]. Additionally, similar attacks have been demonstrated in new-generation technologies such as cloud-computing [67].

Growing interest in side-channel attacks as means to break secure systems can be attributed to at least two reasons. First, benign implementation choices for otherwise secure systems can lead to unanticipated side-channel attacks. Second, side channels often rely on useful implementation features and hence cannot be prevented; e.g., dynamic scheduling of resources at network devices. Unlike covert channels, they do not require the presence of a Trojan Horse or other modifications in the system. These reasons make side-channel attacks more pervasive, and harder to detect-and-counter.

1.1 A Review of Side-channel Attacks

We provide a brief review of well-known side-channel attacks to highlight the threat they pose and motivate the need for precise quantitative analyses of their leakage potential.

1.1.1 Side-channel Attacks in Communication Networks

Traffic-analysis refers to the analysis of network traffic meta data, particularly; packet lengths and timing which allow an adversary to infer private information of communicating parties or breach their communication anonymity. Traffic analysis has increasingly become an alternative to traditional wiretapping which often fails due to the widespread use of end-to-end encryption which ensures content confidentiality. However, the threat to privacy and security caused by traffic-analysis attacks can be as significant. Apart from revealing a user's private data, such as passwords, keys, traffic-analysis can also be used to reveal their communication relationships. Communication relationships provide significant amount of information about a user's identity, behavior, and social milieu, and therefore, are highly sought-after information in law-enforcement community. Numerous side-channel attacks have been developed for these purposes.

For example, inter-packet timings have been used to successfully learn users' SSH password [62]. The SSH protocol transmits each password letter separately as soon as it is entered and encrypted. Inter-packet timing between transmission of two consecutive letter is dominated by the difference in timing of key-presses on the user's keyboard which depends on the relative positions of the two keys. Using the inter-packet timings, the attacker creates a Hidden Markov Model of the key presses and employs it to significantly improve well-known password cracking attacks like dictionary attacks. Packet length and timing characteristics have also been used to identify media-streams [55] and reveal private financial/medical information of web-application users [10]. In addition to passively observing existing timing discrepancies, the attacker can actively create these differences in certain scenarios. Felten and Schneider [19] described an attack where hidden HTML objects from a different website, Website B, are embedded by the attacker in his own website, Website A. If a user has browsed the Website B before browsing WebsiteA, these objects are pre-cached and therefore, the loading time of Website A is considerably less. To prevent leakage of sensitive information through these attacks, the notion of anonymous networking has been developed [51].

Modern anonymous networks are based on an approach developed by David Chaum to

create an anonymous e-mail delivery system. This approach, called *the mix network* [8], is comprised of the following features: 1) encryption of packet content and addresses, 2) division of packets in cells of equal size, 3) use of at least one relay node through which all packets are forwarded, and 4) queuing, delaying, and re-ordering of packets at intermediate nodes. A disadvantage of this approach is the introduction of large packet delay which cannot be tolerated by applications like streaming, VoIP, or browsing. For such applications, specialized anonymous networks have been developed, such as Tor [16], Crowds [53], Web Mixes [3]. Unlike a mix network, these networks do not perform explicit delaying and re-ordering of packets at the intermediate node. Instead, they rely on large traffic and multiple forwarding relays to create an effect similar to mixing. One of the most popular examples of low-latency anonymous networks is Tor [16]. Due to the absence of explicit mixing, low-latency anonymous networks remain susceptible to global adversaries and traffic matching attacks [52]. For VoIP communication, which has an even more stringent delay requirement, no widespread anonymous network exists. Skype is considered to provide anonymity to VoIP users due to its closed-source design. However, it has recently been proven to be vulnerable to traffic-analysis attacks that reveal either the identities of the communicating parties [68] or the identify the network path [50].

Low-latency anonymous networks are particularly vulnerable to side-channel attacks. The sharing of network resources at relay nodes by different traffic streams has an adverse effect on anonymity. Relays in low-latency anonymous networks, such as Tor, buffer packets belonging to different traffic streams and forward them using round-robin or first-come-first-serve scheduling policies. This creates dependencies between traffic load of one stream and the queuing delay of the packets of another stream that shares the relay. This security vulnerability was exploited by Murdoch and Danezis to learn the secret path taken by an anonymous stream in Tor [47]. A similar side-channel attack was launched by Gong *et al.* against home DSL users to learn the websites browsed by them [28].

The side-channel attacks discussed so far exploit weaknesses in scheduling and other policies at the relay nodes, which are part of the network's design. However, side-channel attacks can be used to breach anonymity even if the network is *perfectly private*; i.e. does not reveal user's identity through any information collected inside the network. This attack, named *Private Communication Detection*, exploits side channels present in a communication end-device that reveal the device's busy/idle activity status. For low-latency applications like VoIP, the correlation between busy/idle activity of two communicating parties is high, which allows an attacker to reveal their private communication relationships and call-records [35], [34]. Since the designers of an anonymous network have no control over the end-device, side-channel attacks become feasible even in perfectly anonymous networks.

1.1.2 Timing Attacks against Cryptographic Algorithms

Asymmetric cryptosystems, such as RSA and Diffie-Hellman-Key-Exchange (DH), require repeated computation of modular exponentiations. In the case of RSA (described in Algorithm 1), decryption of a ciphertext c requires the computation of $c^d(mod\ m)$, where d is the private key and m is the publicly-known RSA modulus. In the case of DH (described in Algorithm 2), a participating party secretly selects a value x and computes $g^x(mod\ p)$, where g is the group generator and p is the shared prime. The goal of the attacker is to either learn the secret exponent for these cryptosystems or factorize the modulus in the case of RSA. To achieve this, he is allowed to send chosen ciphertexts and observe the computation time for their exponentiation.

Data: plaintext message:u; large primes: p and q
$m = p.q$;
$\phi(m) = (p-1)(q-1)$;
private key e, such that $1 < e < \phi(m)$ and $gcd(e, \phi(m)) = 1$;
public key d, such that $d.e = 1(mod\ \phi(m))$;
Encryption $c = u^d(mod\ m)$;
Decryption $u = c^e(mod\ m)$;

Algorithm 1: RSA Algorithm

Data: public group generator, g; public prime, p
Alice: select secret $x \in \mathcal{G}$;
Alice: compute $m_a = g^x(mod\ p)$;
Bob: select secret $y \in \mathcal{G}$;
Bob: compute $m_b = g^y(mod\ p)$;
Alice→Bob: m_a;
Bob→Alice: m_b;
Alice: compute $m_b^x(mod\ p) = g^{xy}(mod\ p)$;
Bob: compute $m_a^y(mod\ p) = g^{xy}(mod\ p)$;

Algorithm 2: Diffie-Hellman Key-Exchange between Two Parties

Successful timing attacks have been demonstrated against these cryptosystems in practice for the past two decades [6, 13, 37]. These attacks rely on a side channel created due to the dependence of computation time of a modular exponentiation on the ciphertext, exponent, and modulus. Such discrepancy is caused by two factors:

- Modular exponentiation algorithms, such as *square-and-multiply*, read exponent bits one-at-a-time. Irrespective of the value of the read bit, the temporary variable is multiplied to itself; i.e. squared. However, if the read bit is set, an additional multiplication of the temporary variable and the base is performed, resulting in two multiplication operations. This leads to the operation taking different times for different bit values.

- Modular exponentiation involves repeated modular multiplication which is a computationally-expensive operation. Specific algorithms, in particular, Montgomery Multiplication [46], have been developed to perform it efficiently in hardware. Montgomery Multiplication occasionally requires additional steps for certain values of the multiplicands and modulus. Since the total computation time of an exponentiation is the sum of computation time of each constituent multiplication, it varies with the modulus and base of the exponentiation.

In a typical attack, the adversary first measures the total decryption time for the chosen ciphertext. Using this information, he estimates the most significant bit of the exponent. He computes the time required for processing the ciphertext with this bit offline and subtracts it from the total computation time. The next significant bit is estimated and the process is repeated until sufficient number of bits have been estimated. At this point, the attacker can use number-theoretic relationships between the modulus and exponent to guess the less significant bits. This attack was practically demonstrated by Dhem *et al.* in smart-card devices [13]. Significant optimization using statistical techniques were performed by Schindler [58].

The above attack has the limitation that the attacker is required to know the modulus, M. This is usually the case with most implementations, as RSA modulus and DH prime-modulus are required to be publicly known. However, implementations of RSA that use the Chinese Remainder Theorem(CRT) (Algorithm 3) do not satisfy this requirement [44]. The primary reason for employing CRT in RSA is to increase efficiency of decryption. Under CRT, an exponentiation with large modulus and exponent is replaced with two exponentiations with smaller modulus and exponents.

Data: ciphertext: c; large primes: p and q
$m = p.q$;
$\phi(m) = (p-1)(q-1)$;
$d_p = d(mod\ p)$;
$d_q = d(mod\ q)$;
b_p such that $b_p = 1(mod\ p)$ and $b_p = 0(mod\ q)$;
b_q such that $b_q = 0(mod\ p)$ and $b_q = 1(mod\ q)$;
$c_p = c(mod\ p)$;
$c_q = c(mod\ q)$;
$u_p = c_p^{d_p}(mod\ p)$;
$u_q = c_p^{d_q}(mod\ q)$;
$u = (b_p u_p + b_q u_q)(mod\ m)$;

Algorithm 3: CRT for RSA decryption

Inadvertently, the use of CRT also prevents basic timing attacks. The modulus used to perform an exponentiation in CRT is a prime factor of the RSA modulus and unknown to the attacker. This prevents the attacker from offline computation of intermediate operations. However, Schindler showed that the probability of extra reduction in a Montgomery Multiplication depends on the ciphertext, the exponent bit, and the modulus. Since exponents behave as random bit-sequences , the total number of squarings and multiplications in an exponentiation is independent of the exponent. Thus, the total time can modeled as a normal distribution whose mean and variance are dependent solely on the ciphertext and the modulus. The attacker sends pairs of ciphertexts such that their average computation time differs by a threshold value. The attacker is guaranteed to find the prime modulus (or its multiple) in this range. Successive reductions are performed until the prime modulus can be searched using brute-force. The practical impact of this attack was intensified when Brumley and Boneh demonstrated it remotely against a widely-used cryptographic library OpenSSL[6]. Attacks that use other type of side-channel outputs, like power consumption and acoustic leakage, to break these cryptosystems have been successfully demonstrated as well [21].

Due to the potential of these attacks in breaking widely-used cryptographic algorithms, several countermeasures have also been developed and implemented to thwart them. Timing attacks can be prevented trivially if the decryption oracle always outputs the result after a constant amount of time. However, this constant value must be greater than the worst-case decryption time which makes this approach highly inefficient. Köpf and Dürmuth presented a countermeasure, *input-blinding-and-bucketing*, where the output of a decryption is revealed during pre-specified windows of time [39]. This countermeasure hinders timing attacks as more ciphertext require same amount of decryption time and is more efficient than the constant time approach.

One of the most popular countermeasure against timing attacks, *exponent blinding*, was proposed by Kocher [37]. Here, a random salt, r, is added to the exponent d. First, an exponentiation is performed using $(r + d)$ as the exponent. An additional exponentiation is performed using r as the exponent. Division of the results of these exponentiations yields the original outcome. Since the exponent used for decrypting each ciphertext is random, the attacker cannot estimate it. Another popular countermeasure, *caching*, uses memory to thwart timing attacks without performance penalties. The results of multiplication of numerous pairs of multiplicands are pre-computed and stored in memory. When such a pair is encountered during an exponentiation, the algorithm simply performs a constant-time memory lookup to retrieve the output. This approach reduces the total number of multiplications performed live during an exponentiation, reducing timing discrepancy. Despite these measures, timing attacks remain one of the biggest threats against modern cryptosystems and newer development need to be resilient to them.

These attacks demonstrate the negative impact side channels can have on a system's security. Detection and mitigation of these attacks is crucial to develop trustworthy and dependable systems. Although there are several works that demonstrate side-channel attacks in different setups, very few attempt to provide precise quantitative analyses that are applicable beyond specific attacks. In the next section, we discuss the advantages, and limitations of past quantitative approaches to side-channel analyses.

1.2 Quantitative Analysis of Information Leakage

Primary focus of past side-channel research is on detection, demonstration, and mitigation of specific attacks. However, demonstration of specific attack techniques on a system does not give much insight on other (and potentially all) possible attacks that use the same vulnerability. Furthermore, they do not provide insights on whether the attacker uses his resources with most efficiency. For example, side-channel attacks against Tor and DSL routers remain proof-of-concepts because the bandwidth resources required to launch attacks in real networks is enormous[18].

In the absence of quantitative analyses, security guarantees of countermeasures cannot be established. This leads to the use of informal countermeasures which may lead to a false sense of security. For example, exponent blinding was traditionally believed to prevent all timing attacks and was adopted into most standard cryptographic libraries. Recent results, however, show that timing attacks may still be possible [59]. Similarly, use of firewalls and anonymous networks, traditionally accepted as countermeasures to traffic analyses, have been shown to not prevent side-channel attacks in communication networks [61],[47],[28].

Quantitative approaches can contribute to all three dimensions of side-channel research: detection, demonstration, and mitigation of attacks. For example, one of the most powerful practical attacks against RSA developed by Brumley and Boneh [6], relied on the theoretical basis developed Schindler [58]. Later in this thesis, we demonstrate how quantitative models enable the development of optimal attack strategies against packet schedulers for a given attack budget. These optimal strategies achieve up to 1300% gain in information leakage than proof-of-concept strategies [27]. Quantitative analyses can also provide system designers the knowledge of system parameters that leak the least amount of information. Several design choices can also benefit from such analyses. For example, in this thesis we compute the leakage of

modular exponentiation based cryptographic algorithms and study the impact of Montgomery reduction parameter on the leakage. This allows us to identify the value of the parameter that leaks least information.

Quantitative analyses can also help system designers develop provably-secure countermeasures. Ghaderi and Srikant developed optimal mixing strategies for preserving anonymity of users [22]. Mathur and Trappe [43] performed analyzed randomization-based countermeasures that prevent anonymity leakage through packet length and timing characteristics. Kadloor *et al.* developed mathematical models for a shared packet scheduler to develop privacy-preserving packet scheduling schemes [36]. Similarly, Köpf and Dürmuth proposed a timing bucket-based countermeasure to timing attacks against RSA and quantify the security provided by it [39].

The first step towards a quantitative analysis is to develop an appropriate model for the system and select suitable metrics that measure relevant performance parameters. Information-theoretic metrics have been favored in most existing side-channel analyses. The primary reason for this choice is that a number of side-channels can be modeled as stochastic systems. That is, the statistical relationship between side-channel inputs and outputs can be represented as a conditional probability distribution of outputs given the inputs. The behavior of user's inputs is also favorably modeled as a random process; e.g., cryptographic keys as uniformly-random binary strings and packet arrivals as Poisson/Bernoulli-distributed sequences. For such models, information theory has a rich set of results and quantities like entropy, mutual-information, capacity, and error-exponents [11], can be used as metrics for security properties like confidentiality and anonymity.

Some of the past works on side channels perform information-theoretic analysis of leakage in both cryptographic algorithms and communication networks. For cryptosystems, Gierlichs *et al.* [23] abstracted a side channel as a mapping between the cryptographic key K, the side channel inputs $x_1, x_2, \cdots, x_n$ and the corresponding side channel outputs $o_1, o_2, \cdots, o_n$. They empirically computed the distribution $P_K(O|X)$ and used empirical mutual-information as a classifier; i.e. the key $\hat{K}$ was the estimate if it maximized the empirical mutual information $\max_K H_k(X) - H_K(X|O)$. Köpf and Basin created an information-theoretic model for adaptive side-channel attacks on a generic cryptosystem and obtained upper-bounds on the remaining key-entropy after n uses of the channel [38]. Köpf and Dürmuth used method-of-types results to show that the number of bits of the cryptographic key revealed by a side-channel attack is upper-bounded by $|O| \log_2(n+1)$, where $|O|$ is the number of possible side channel outputs and n is the number of observations [39][40]. Mizuno *et al.* used channel capacity of analog communication channels; i.e. $\frac{1}{2}\sqrt{1 + \frac{signal-power}{noise-power}}$, as a metric for attacks that employ power consumption analysis [45]. Demme *et al.* used correlation between the user's behavior and the attacker's observation to develop a metric called the *side-channel vulnerability factor* [12].

In communication networks, information-theoretic metrics have traditionally been used as anonymity measures. Berthold *et al.* [4] measured the anonymity of a communication pair as the size of the *anonymity set*, which is the set of all pairs that could possibly communicate. However, each communicating pair may not be equally likely to generate a specific communication activity. For such scenarios, Serjantov and Danezis [60] provided an alternate metric in terms of the *Shannon entropy* of the probability distribution over the anonymity set. Under the chosen metric, anonymity leakage is defined as the reduction in anonymity due to the attacker's observations. The reduction in anonymity is sometimes normalized with the a-priori anonymity to account for different starting conditions [14]. Based on these met-

rics, several works have quantitatively analyzed anonymity systems. Ghaderi and Srikant [22] quantified the anonymity of a mix node under attacker's observations and identified the strategies that maximize anonymity under a given delay constraint. Mathur and Trappe [43] performed an information-theoretic analysis of randomization-based countermeasures that prevent anonymity leakage through packet length and timing characteristics. Kesdogan *et al.* [17] explored the notion of probabilistic anonymity for anonymous networks and developed a countermeasure, *stop-and-go mixing*. Chatzikokolakis *et al.* modeled anonymity protocols as noisy channels and used channel capacity as an anonymity metric [7]. Gong *et al.*[27] developed a mathematical model of shared packet schedulers and computed the information leakage in terms of the mutual information between user's packet arrival rate and attacker's probing rate.

Despite significant effort on quantitative analyses of side-channel attacks, a number of limitations to current approaches exist which reduce their impact on practical system design. Specifically,

- *Choice of non-uniform quantitative metrics:* There is significant diversity in the choice of metrics used in the literature. Metrics used for information leakage range from reduction in entropy to normalized mutual-information and capacity. Other metrics like side-channel vulnerability factor, rely on correlation as opposed to entropy. The association between these metrics and real-world performance measures, like accuracy of information learned, has not been established. Additionally, the diversity of metrics makes it difficult to develop relationships among them and therefore, comparative analysis becomes challenging.

- *Choice of specific attack strategies:* Even when a detailed model of the side channel is available, several analyses are limited to specific and often simplistic attack strategies. Adaptive strategies, where the attacker uses past information to decide next inputs, are generally ignored even when adaptive attacks, which are demonstrably stronger, are feasible. As we report later in the thesis, adaptive attacks can cause significantly more information leakage for the same resource budget. Thus, even under suitable metrics, most analyses do not measure the worst-case leakage of the system.

The results reported in this thesis remove these limitations by providing a general side-channel model which can be used to quantitatively analyze a wide variety of side channels. Under this model, we define three metrics: 1) *capacity*, 2) *reliability rate*, and 3) *leakage*. We show that *capacity* is an ill-suited metric for side-channel as it cannot guarantee accuracy of retrieved information. Instead, we propose the use of *reliability rate* and *leakage* which measure the accuracy of leaked information and rate of information leakage, respectively. We define notions of security under both metrics and show that these notions are not equivalent to each other.

1.3 Analysis of Information-leakage Metrics: *Capacity*, *Reliability Rate*, and *Leakage*

In principle, side channels are statistical relationships between the user's secret, attacker's inputs, and side-channel outputs. This relationship can be represented as a probability distribution on side-channel outputs conditioned on side-channel inputs. A generalized side-channel model can be developed by using this description of side channels.

1.3.1 A General Side Channel Model

A general side channel can be modeled as a discrete-time, two input-single-output system, as illustrated in Figure1.1. In every time-slot, the attacker and user issue one input each. The side channel produces one output that depends on the inputs and previous outputs. The attacker observes this output and uses his observations to estimate of the user's inputs.

User's input process: Depending on the scenario, the user's input can be a single value; e.g., a cryptographic key, or a sequence of values; e.g., a stream of packets in a network [16]. For the first case, the user's input is represented by a single value $d \in \mathcal{D}$, whereas, for the latter case, the i^{th} input of the user is represented by $d_i \in \mathcal{D}$.

Usage Mapping: The two scenarios, where the user issues a single input, d, or a sequence of inputs, d^n, are not conceptually distant from each other. The user can make a singular choice $h \in \mathcal{H}$ about his true secret; e.g, a website, and the choice of h determines the d^n; e.g., a sequence of packets, that is input to the side channel through a mapping $\mathcal{U} : h \rightarrow d^n$. For example, in shared packet schedulers, the user may be assumed to generate a sequence of packets, d^n, through a random process; e.g., Bernoulli [27], or to select a website h which then determines the sequence of packets d^n [26]. In the first case, the attacker would aim to learn the sequence, d^n, while the object-of-interest in the later case is h. The mapping $\mathcal{U}$ is named *usage mapping* because it determines how the system is used for a specific choice made by the user, for example, the packet arrival pattern of a chosen website. This distinction is important in distinguishing *reliability rate* and *leakage* metrics.

Attacker's strategy: The attacker strategy is specified by the choice of his inputs. Let $x_i \in \mathcal{X}$ represents the attacker's i^{th} input to the system. For a general case, x_i might be chosen as an adaptive function of all the previous inputs the attacker has issued to the system and corresponding outputs; i.e. $x_i = f(x^{i-1}, y^{i-1})$. After issuing n inputs to the system, x^n, and observing the corresponding side-channel outputs, y^n, the attacker applies an estimator g to produce an estimate $\hat{d}$ of the user's secret. The tuple (f, g) collectively specify the attacker's strategy. If the attacker's i^{th} input is independent of past observations, then the strategy is said to be non-adaptive.

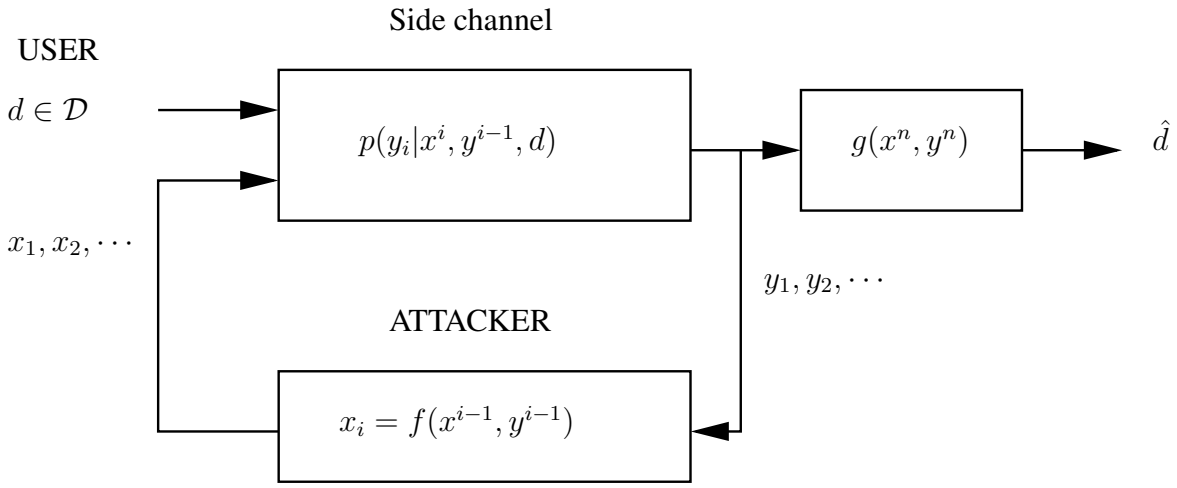


Figure 1.1: A generic model for side channel

Side-channel model: Let $y_i \in \mathcal{Y}$ be the i^{th} output of the side channel. Then, in the most general case, y_i is dependent on all the previous inputs x^i, d^i, and outputs y^{i-1}. The stochastic

relationship, represented as a probability distribution $P(y_i|x^i, d^i, y^{i-1})$, specifies the side channel in entirety. The side channel is assumed to be non-anticipatory or causal; i.e. outputs do not depends on future inputs. The side channel is called memory-less if y_i is statistically independent of x^{i-1}, d^{i-1}, and y^{i-1}, given x_i and d_i. In cases where the user only makes one input; e.g., a secret key, to the system, the side channel is described by the distribution $P(y_i|x^i, y^{i-1}, d)$. The goal of the attacker is to learn the user's input, d or d^n (eventually h), depending on the setup.

Definition 1. *The probability of error P_e of an attacker under strategy $f(), g()$ is defined as:*

$$P_e \equiv P(g(x^n, y^n) \neq d)$$

P_e serves as a measure of success for any attack strategy as it quantifies reliability of information learned under an attack strategy.

We note that P_e may not be a suitable metric in scenarios where the attacker can tolerate some distortion in the information learned. A different metric, such as average Hamming or Euclidean distance, must be used in such scenario. For the scope of this work, we limit the discussion to P_e. With a general stochastic model for side channels, we proceed with our discussion on the right metrics to quantify information leakage.

1.3.2 Comparison of Quantitative Metrics of Information Leakage

Intuitively, an information flows, such as a side channel, is a statistical relationship between the inputs and outputs of the flow that allow estimation of the input by observing the output. Other examples of information flows are classical transmission channels or covert channels which can be used by a Trojan Horse to cross system's access control boundaries. For any information flow, a metric of interest is the size of the flow; i.e. number of bits transferred/leaked per channel use. At the same time, it is also important to ensure that the information transferred/leaked is accurate (reliable) and that the metric can be easily generalized to analyze different setups that use the same channel.

For transmission channels and covert channels, the notion of capacity fulfills all the requirements [25]. It measures the maximum rate (bits per channel use) of information transfer while satisfying reliability constraints. *Capacity* also bounds the rate achieved by any communication system which uses the underlying channel. It serves as a performance benchmark for all encoding-decoding schemes, thus providing a general analysis for the given channel. While *capacity* may be an attractive metric for analyzing side channels, to date, no work has been able to demonstrate a direct relation between side-channel *capacity* and vanishingly low P_e. Discouragingly, as we show later in this section, this association is impossible for side channels.

This motivates us to propose the use of two metrics: *reliability rate* and *leakage*, that measure accuracy of information and leakage rate, separately. Both metrics are applied to distinct side-channel setups, *reliability rate* for a single user input and *leakage* for a sequence of inputs from the user. *Reliability rate* is measured as the optimal error-exponent of the attacker in a hypothesis-testing framework, where he estimates the underlying user input $d \in \mathcal{D}$. A positive *reliability rate* ensures that P_e approaches zero as the number of attacker's inputs approaches ∞. In contrast, *leakage* is measured in terms of the asymptotic mutual-information

rate between the user's input sequence and the attacker's input-output sequence. Since *leakage* measurement is independent of the usage mapping, the analysis applies to all mappings, providing a new metric that is distinct from the *reliability rate*. Table 1.1 summarizes how each metric fares along the directions of reliability and generality. We now define each metric formally, which enables us to provide rigor to intuition.

Metric	Reliability	Generality
Capacity	✓	✓
Reliability rate	✓	✗
Leakage	✗	✓

Table 1.1: Performance of different metrics on two aspects: reliability of information and generalization over all security parameters

Capacity

We first review the definition of channel capacity for classical transmission channels and use it to develop a similar definition for side-channels. In the case of classical transmission channels, channel capacity measures the maximum number of bits that can be reliably transmitted through the channel. The transmitter chooses a message, $M \in \mathcal{M}$, that needs to be transmitted over a discrete memory-less channel (DMC). The channel takes inputs $x_i \in \mathcal{X}$ and produces corresponding outputs $y_i \in \mathcal{Y}$ at the receiver. The channel is specified by the stochastic mapping $P(y_i|x_i)$. Given the message, M, the encoder chooses an n-length code word, x^n through an encoding function $f : \mathcal{M} \to \mathcal{X}^n$. The receiver receives a series of outputs, y^n, produced by the channel. Using y^n, the receiver estimates the transmitted message $\hat{M}$ using a decoding function, $g : \mathcal{Y}^n \to \mathcal{M}$. A error is made if $\hat{M} \neq M$. A rate $\mathcal{R}$ is said to be admissible, if there exist a number n, and encoding function, f, and a decoding function, g such that

$$\lim_{n\to\infty} P(\hat{M} \neq M) = 0 \text{ and } \lim_{n\to\infty} \frac{\log|\mathcal{M}|}{n} \geq \mathcal{R}$$

The channel capacity is defined as the maximum achievable rate; i.e.

$$\mathcal{C} = \sup_{\mathcal{R} \text{ is admissible}} \mathcal{R}$$

For a DMC specified by the probability distribution $P(y|x)$,

$$C = \max_{p(x)} I(X;Y).$$

We formally define the *capacity* of a general side channel model along the same lines.

Definition 2. *A side-channel rate, $\mathcal{R}$, is said to be admissible if there exists an encoding $f()$ and an estimator $g()$ such that*

$$(1) \qquad \lim_{N\to\infty} P_e = 0$$

$$(2) \qquad \lim_{N\to\infty} \frac{\log m}{N} \geq \mathcal{R}$$

Definition 3. *The* capacity*, $\mathcal{C}$, of the side channel is defined as the supremum of all admissible rates, i.e.*

$$\mathcal{C} = \sup_{\mathcal{R} \text{ is admissible}} \mathcal{R}$$

This definition essentially provides a measure for the maximum asymptotic rate at which the user's information can be learned by the attacker while ensuring reliability of information learned. However, this definition is ill-suited for a general side-channel for the following reasons.

- The side-channel probability transition function $P(y_i|x_i, d)$ does not scale with the increase in the user's input space. For example, the side-channel description for 256-bit RSA is completely different that for 512-bit RSA. This implies that number of inputs required to achieve a given P_e need not scale exponentially with the size of the secret.
- Alternatively, if the user's input space is fixed, then the number of bits that the attacker needs to learn is finite and therefore, the asymptotic *capacity* is zero.
- The user's inputs to the side channel depend on the usage mapping and therefore, cannot be guaranteed to have a minimum separation required to achieve low probability of error.
- Existence of *capacity* relies on the existence of sequences (code words) that lead to disjoint output sequences. The encoder selects one of such sequences to transmit a specific message. This, however, requires the knowledge of the message that needs to be transmitted. In the case of side-channel, the choice of the secret is made by the user and is unknown to the attacker, prohibiting him from selecting optimal code words.

In the absence of a viable definition of side-channel *capacity*, we formulate alternate metrics: *reliability rate* and *leakage* to measure accuracy and rate of information leakage.

Reliability rate

Irrespective of the form of information of the user, a side channel attack can be modeled as a multi-hypothesis testing problem. The user's input to the side channel is either a single secret $d \in \mathcal{D}$ or a sequence of inputs, d^n which may be specified by his secret h through a usage mapping, $\mathcal{U} : h \rightarrow d^n$. The attacker issues n inputs, x^n to the system (adaptively or non-adaptively) and observes the corresponding n-output vector y^n. Using x^n and y^n, the attacker identifies the underlying hypothesis $\hat{d}$ (or $\hat{h}$). The attacker's probability of error w.r.t. i^{th} hypothesis is given by

$$P_e(i) = P[\hat{d} \neq d_i | d_i]$$

The average probability of error P_e is given by,

$$P_e = \sum_{d_i} P(d_i) P_e(i).$$

The *reliability rate* of an attack strategy, specified by the input distribution $f()$ and the classifier $g()$, is defined as,

$$reliabilityrate = \lim_{n \rightarrow \infty} \frac{-\log P_e}{n}$$

Reliability rate essentially measures the asymptotic exponential-rate at which the probability of error decreases with the number of samples n. The higher the *reliability rate*, the faster the probability of error reduces which implies higher accuracy. A positive *reliability rate* ensures that the P_e goes to 0 with increasing n. This metric is very effective at comparing different attack strategies for the same setup. However, the analysis of *reliability rate* varies for different $\mathcal{D}$, prior distributions on the hypothesis, or different usage mappings and is not general. To provide generality of analysis, we next define the *leakage* metric.

Leakage

Leakage is defined for the scenario when the user issues a sequence of inputs, d^n. *Leakage* attempts to measure the asymptotic mutual information rate between the user's input and the attacker's observations, x^n and y^n. This is is defined as follows.

Definition 1. Leakage *of a side channel, $\mathcal{L}$, for an attack strategy $p(y_n|y^n, w^n)$, is defined as:*

$$\mathcal{L} = \lim_{n\to\infty} 1 - \frac{H(D^n||X^n, Y^n)}{H(D^n||X^n)}$$

where, $H(\ ||\ )$ represents causally-conditioned entropy [42]. *Leakage* has certain favorable properties:

- $\mathcal{L} \in (0, 1)$, where *leakage* of 0 implies no information leakage but *leakage* of 1 implies total information leakage
- Since *leakage* measures the mutual information between the direct inputs and outputs of the side channel, it is independent of usage mappings. This has an advantage over *reliability rate* which is defined for a specific usage mapping and therefore, more general.
- While *leakage* cannot be used to provide strong bounds on P_e, it can be used to provide bounds on other performance metrics, such as average distortion.

Although *reliability rate* and *leakage* complement each other, in terms of computing the accuracy and the rate of information leakage, these metrics are not equivalent. Since *leakage* quantifies the amount of information leaked about final side-channel inputs from the user, security under this metric may be considered *intuitively* stronger. Notions of semantic security under each criteria can be defined as,

Definition 2. *A system is said to be secure under* leakage *criteria; i.e. $L - secure$, if the side-channel* leakage *is 0. Similarly, a system is said to be secure under the* reliability rate *criteria; i.e. $R - secure$, if the* reliability rate *is 0.*

Then, we have

Theorem 1. *1) The security of a system under the* leakage *criteria does not imply the security of system under the* reliability rate *criteria; i.e.*

$$L - secure \nRightarrow R - secure$$

2) The security of a system under the reliability rate *criteria does not imply the security of system under the* leakage *criteria; i.e.*

$$R - secure \nRightarrow L - secure$$

That is, neither of these notions of security can guarantee the other.

Proof. We create counterexamples when both of these relationships are true.
1) Consider a channel in which the output of the channel, y_i is noiseless for $i = \{0, \cdots, \log_2 |\mathcal{H}|\}$ and completely random and independent of the inputs for $i > \log_2 |\mathcal{H}|$. In that case, the user might leak-out the hypothesis, h during the first $\log_2 |\mathcal{H}|$ by simply leaking the secret. In this case, since the remaining y are independent of the inputs, the limiting mutual information-rate will be 0. Thus, the system will be $L - secure$ but not $R - secure$.

2) To show the lack of a converse, we provide a different counterexample that where the system is $R - secure$ but not $L - secure$. Consider the example of side channel in packet schedulers. Here, the *leakage* of the system measures the mutual information between the attacker's observations and the user's packet arrival pattern. Assume that the *leakage* of the system is 1; i.e. the attacker can learn the user's packet arrival pattern with complete certainty. Even in this case, if all websites map to the same the packet arrival pattern; e.g., a constant bit-rate traffic stream, the attacker cannot distinguish between any two website by performing a side channel attack; i.e. the system is $R - secure$ but not $L - secure$. This shows that $R - secure \nRightarrow L - secure$, which completes the proof. □

This theorem shows that these two criteria for security against side-channel attacks are different from each other and therefore, the computation of each of them provides different insights on the security of systems. Using this analysis as a foundation, the thesis makes the following contributions.

1.4 Contributions

The first contribution of the thesis is the analysis of information leakage metrics presented in the previous section. We use the *reliability rate* and *leakage metrics* to analyze three different side channels in : a) communication end-devices, b) network components, and c) cryptographic algorithms.

Side channels in communication end-points

Side channels present in a communication end-point device reveal its busy/idle activity status. Correlation in such activity among multiple parties can leak private communication relationships represented by *call-records*. We develop a mathematical model for Private Communication Detection of two parties where the attacker sends periodically probes each party and observes their busy/idle status over a period of time. He uses this information to learn call-records and breach caller-callee anonymity. For this model,

- We compute the reliability rate of the attacker in detecting private communication between these parties and analyze its relationship with communication parameters and probing rate of the attacker.

- For two communicating parties, we compute the leakage of their call-record information achieved by PCD. We analyze the impact of observation noise on anonymity leakage and compute the reduction in leakage in terms of the channel capacity of noisy-channel.
- We develop resource-randomization based countermeasures against PCD. These countermeasure work by adding artificial noise in the attacker's observations. The anonymity gain of such countermeasures is defined in terms of the normalized reduction in anonymity leakage of the system. We show that our countermeasures can potentially thwart PCD completely.

Side channels in network components

Side channel present in network components, such as packet schedulers, allow an attacker to learn traffic patterns of private traffic streams. Knowledge of these patterns can be used to identify the source or the path of traffic stream, breaching traffic anonymity. We use the theoretical model of first-come-first-serve packet scheduler, developed by Gong *et al.* [27], to identify *optimal attack strategies* for leakage of traffic patterns. Optimal strategies make efficient use of the attacker's bandwidth resources and allow him to perform large-scale attacks. In each of the following cases, optimal strategies are identified by solving linear programs which make it easy for an attacker with moderate capabilities to use them. Specifically,

- We discover optimal non-adaptive attack strategies for a given attack bandwidth and demonstrate upto 1000% gain in leakage compared to geometric probing of [27].
- We develop a new leakage metric to analyze adaptive strategies and demonstrate upto 30% increase in leakage compared to optimal non-adaptive strategies, highlighting the importance of analyzing adaptive side-channel attack strategies.
- We identify optimal real-world strategies where the attacker has a limited view of past outputs and show that they achieve higher leakage compared to non-adaptive strategies for the same attack bandwidth.

Side channels in cryptographic algorithms

Decryption times of chosen ciphertexts allow an attacker to *learn the secret key* or *modulus* used in a modular exponentiation-based cryptosystems. For quantifying information leakage in cryptographic algorithms using our metrics, we employ Schindler's stochastic model for computation times of a modular exponentiation [58]. In particular,

- We compute the optimal reliability rate of an attacker in estimating secret prime modulus for RSA with CRT.
- We develop a novel asymptotic model for the timing side channel in Montgomery Multiplication and show that the leakage of the algorithm computed under this model provides an upper bound for the side-channel leakage of any cryptosystem that uses the Montgomery Multiplication routine.
- We finally analyze two well-known countermeasures to timing attacks: exponent blinding and caching. We compute the reduction in leakage under each countermeasure and identify the conditions under which one outperforms the other.

In addition to analyzing side-channel attacks and countermeasures under these diverse systems, this thesis also provides a method for quantitative analysis of side-channel attacks in other systems.

Chapter 2

Side Channels in Communication End-point Devices

Communication records often reveal private relationships between two communicating parties. For example, they capture interaction frequency over time, evidence of recent interaction, communication reciprocity, and the existence of at least one mutual friend that links the two parties. These parameters provide a fairly accurate indication of tie strength between two parties [24]. As a consequence, communication-record analysis has been one of the key tools used by analysts to discover the social milieu of targeted individuals or groups. Naturally, access to communication records is restricted by law in many countries and carefully controlled by service providers who collect them [29].

Privacy concerns raised by wholesale collection of VoIP and other call records have led to question of whether collection of such records could be thwarted by the use of *private networks*. These networks would not merely provide the confidentiality of the call content. Equally importantly, they could also provide both flow anonymity and user pseudonymity properties, which would make wholesale collection of call records challenging. However, even if private networks could support VoIP calls in the future, they would still be vulnerable to a side-channel attack, known as *private communication detection (PCD)* [35]. In a PCD attack, an analyst, henceforth called the *attacker*, first detects the busy/idle activity status of a target by exploiting resource-contention side channels present in a target's VoIP device. He then correlates this information for multiple targets over a period of time and discovers their pairwise call records. Such attacks have been demonstrated in a number of common communication technologies including VoIP, Wi-Fi, 3G, and instant messaging [34] [33]. More importantly, they can be launched remotely, at low cost, and do *not* require direct manipulation of the targets' communication resources. However, they require that an attacker has the ability to send/receive messages to/from the targeted communication devices without detection. Clearly, the mere correlation of the activity of different targets may not necessarily imply the existence of a relationship between them. For example, if the attacker's measurement of a target's busy/idle activity status is noisy, the correlation between the observed activities cannot be relied upon. Furthermore, in the presence of multiple users in the network, two targets may be busy at the same time while talking to other communication partners and not with each other. Hence, simplistic analyses might indicate the existence of a relationship when none exists. Accurate analyses become important both for privacy advocates, who want to quantify the amount of anonymity leakage caused by PCD and the efficacy of countermeasures, and for the call-record

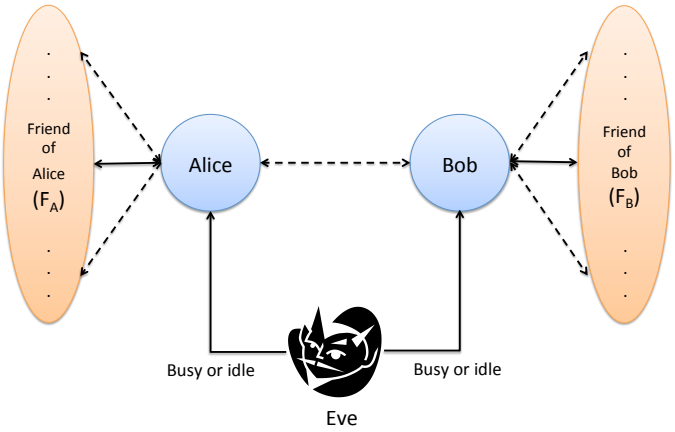


Figure 2.1: Network model

collector (i.e., the attacker), who may have to rely on PCD when no other collection means are available; e.g., in a foreign jurisdiction.

In this chapter, we study the efficacy of PCD for a two-target scenario illustrated in Figure 2.1. Here an attacker, Eve, periodically probes two targets, Alice and Bob, aiming to collect enough activity logs and discover whether a relationship exists, with demonstrable accuracy. We develop a mathematical model to represent the calling behavior of the two targets and the probing strategy of the attacker. Under this model, we make the following contributions.

- **Reliability rate of the attacker in estimating communication relationships:** We provide upper bounds on the probability of an attacker and the reliability rate in accurately classifying the communication relationship between the two targets (i.e., as existent or non-existent). We analyze its relationship with parameters like the number of samples collected, the probe rate, and call parameters.

- **Quantitative analysis of the anonymity leakage:** Once the communication relationship between the two targets is accurately established, the attacker aims to learn their communication details; e.g., the time and length of each conversation between them. We compute the leakage of call-record information due to the knowledge of activity-logs under the definition presented in Chapter 1.

- **Countermeasures and their efficacy:** We study the efficacy of practical countermeasures, such as resource randomization and firewalls, which thwart PCD attacks in a quantifiable manner. Using our leakage model, we measure the efficacy of a countermeasure as the reduction in the anonymity leakage. Our analysis shows that resource randomization outperforms the use of firewalls and has the potential to completely thwart PCD by introducing noise in the adversary's side channel. In some cases, however, the use of randomization is limited due to system usability constraints.

2.1 System Description

In this section, we present the employed network and call arrival-service model for analyzing PCD in a two-target scenario and the reasoning behind the assumptions made for the analysis. First, we specify the network model and assumptions.

2.1.1 Network Model

We consider a simple communication network consisting of two targets: Alice and Bob, an attacker Eve, and third-parties: F_A and F_B, as shown in Figure 2.1. Eve is aware of the existence of a communication relationship between Alice-F_A and Bob-F_B but does not have any a-priori information about the existence of the communication relationship between Alice and Bob, either existent or non-existent. Eve probes both Alice and Bob but is not capable of probing either F_A or F_B. While, this model assumes that both Alice and Bob have only one third-party friend each, under appropriate modeling of call arrival-service, the existence of multiple third-party friends can be abstracted by that of a single entity, F_A or F_B. For example, let Alice have two third-party friends: Carol and David. If Alice speaks to Carol for 5 minutes in an hour over multiple calls and to David for 15 minutes in an hour, then they both may be represented by one friend, with whom Alice speaks for 20 minutes in an hour. This abstraction is warranted because Eve is not interested in the communication relationship or call records between Alice and her third-party friends, and also explains Eve's inability to probe F_A or F_B as they may not be unique individuals. In Section 2.2, we provide the call arrival-service model that enables us to use this abstraction.
Remark 1: Realistically, Alice and Bob may have common friends; i.e. $F_A \cap F_B \neq \emptyset$, but we assume that Alice and Bob do not talk to a mutual friend simultaneously. This assumption is justified as we are only interested in the pair-wise communication relationships of the targets and not in teleconferencing over multiple parties.
Remark 2: The attacker is only required to know the aggregate communication parameters; i.e. call arrival-service rates between Alice, Bob, and their third-party friends. It is not required to possess knowledge of either the identities or number of third-party friends of Alice and Bob. The attacker's lack of knowledge of the identities of the third-party friends is another reason for his inability to probe them. This model arises in many real-world scenarios, such as,

- **Detecting communication relationships across privacy-preserving jurisdictions**: Consider two countries that authorize their law enforcement agencies to perform bulk domestic call records collection, and assume that these agencies cooperate to detect communication relationships between selected individuals across their national border. However, privacy protection laws may prevent exchanging call records across national borders, but allow the sharing of aggregate communication parameters, since they do not reveal individuals' identities or call patterns. Our PCD model enables the detection of call relationships between such individuals without requiring exchanges of private foreign call records.

- **Detecting communication relationships using past call records**: Legally-authorized, bulk call record collection is typically restricted in time; e.g., six months. Suppose that retention of derived aggregate communication parameters beyond the temporary call record

authorization is not prevented, since it does not reveal individuals' identities or call patterns. Under these circumstances,our PCD model enables the detection of communication relationship between two selected individuals based on past aggregate communication parameters, since PCD can be performed without any legal restrictions; i.e., access to public signals cannot be prevented.

The above examples show that even when individuals' identities or call patterns are not revealed by innocuous aggregate communication parameters, PCD can in fact lead to call record collection. Next, we specify the call arrival-service model and assumptions.

2.1.2 Call Arrival-Service Model

Call arrival process: Call arrivals between any two communicating parties follow the Poisson call arrival process with a corresponding arrival rate. The Poisson arrival process has been used extensively to model call-arrivals in telephony systems in the literature. Empirical studies have also verified that the Poisson distribution models call arrivals at a real telecommunication node, such as a telephone exchange, accurately [65]. Additionally, we assume that all communicating parties are non-colluding and no ancillary information about the social graph of third-parties is available. Under these conditions, calling behavior of different parties is independent of each other.

Call service model: The service time for each call, or the call duration, is assumed to follow the i.i.d exponential distribution with an appropriate rate parameter.

Time in the system is discretized into blocks of length "Δ" seconds, which serves as the fundamental unit of time in the system. Δ is assumed to be sufficiently small so that the device can only undergo one transition per time-slot.

A simple example: Consider a device A that only has a single communication partner B. Calls arrive on this line following the Poisson arrival process with rate λ. Thus, the inter-arrival time between consecutive calls follows an i.i.d. exponential distribution. Also, call duration follows the exponential distribution with parameter μ. In any given time-slot, A can be on a call with B (busy) or not (idle). If A is idle in the current time-slot, then it remains idle in the next time-slot if there is no call arrival in the next Δ seconds; i.e. with probability $e^{-\lambda\Delta}$. A transitions to busy in the next time-slot with probability $1 - e^{-\lambda\Delta}$. Similarly, if A is busy in the current time-slot, then it remains busy in the next time-slot if the remaining call duration is greater than Δ seconds; i.e. with probability $e^{-\mu\Delta}$. A transitions from busy to idle in the next time-slot with probability $1-e^{-\mu\Delta}$. Due to the memorylessness of the call service time and call inter-arrival time distribution (both exponential), the calling behavior of A can be represented as a stationary first-order Markov chain, as depicted in Figure 2.2.

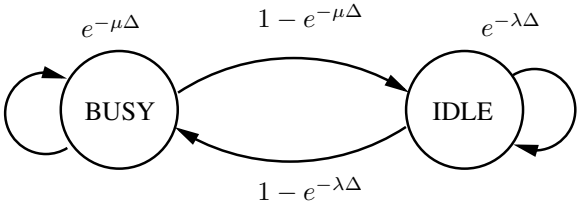


Figure 2.2: Transition diagram for a single device

Another salient feature of the Poisson arrival model is that it satisfies the condition mentioned in Section 3.1, i.e. capturing the effect of multiple third-party friends in a single third-party with appropriate modifications in the rate parameters. Specifically, if Alice has two friends Carol and David, and Alice speaks to Carol with a rate λ_C and to David with a rate λ_D, then the probability that Alice remains idle in the next time-slot equals the probability of no call arrivals from either Carol or David: i.e. $e^{-\lambda_C \Delta} e^{-\lambda_D \Delta} = e^{-(\lambda_C + \lambda_D)\Delta}$ (due to independence of call-arrivals from different parties). This makes the existence of Carol and David equivalent to a single third party friend who speaks to Alice with a call rate $\lambda_C + \lambda_D$. Similar abstraction can be done for call duration.

Now, we extend this model to represent the calling behavior of Alice and Bob in two scenarios: a) when they do not have a communication relationship with each other and b) when they have a communication relationship with each other. In scenario a), Alice and Bob have social relationships with F_A and F_B respectively but not with each other. Therefore, their isolated individual behavior can be represented by the model in Figure 2.2 albeit with different parameters (λ_A, μ_A) and (λ_B, μ_B). In the absence of a communication relationship between them, the behavior of Alice and Bob is independent of each other and therefore, the collective transition probabilities are the product of respective individual transition probabilities. Alice and Bob can have four possible states: (Alice is idle, Bob is idle), (Alice is idle, Bob communicates with F_B), (Alice communicates with F_A, Bob is idle), and (Alice communicated with F_A, Bob communicates with F_B) which are represented as (0,0), (0,1), (1,0), and (1,1) respectively. The collective state transition diagram is shown in Figure 2.3.

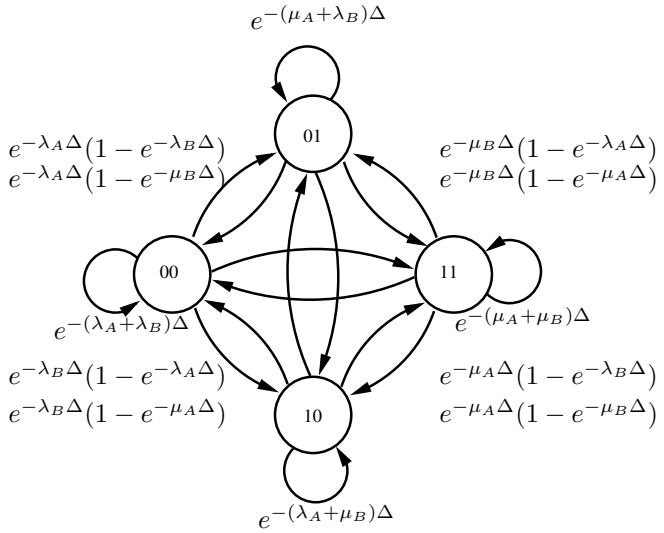


Figure 2.3: Joint transition diagram when no relationship exists between Alice and Bob

The stationary distribution (π) of this Markov chain can be calculated as:

$$\begin{aligned}
\pi(00) &= \frac{\left(1-e^{-\mu_A\Delta}\right)\left(1-e^{-\mu_B\Delta}\right)}{\left(2-e^{-\lambda_A\Delta}-e^{-\mu_A\Delta}\right)\left(2-e^{-\lambda_B\Delta}-e^{-\mu_B\Delta}\right)} \\
\pi(01) &= \frac{\left(1-e^{-\mu_A\Delta}\right)\left(1-e^{-\lambda_B\Delta}\right)}{\left(2-e^{-\lambda_A\Delta}-e^{-\mu_A\Delta}\right)\left(2-e^{-\lambda_B\Delta}-e^{-\mu_B\Delta}\right)} \\
\pi(10) &= \frac{\left(1-e^{-\lambda_A\Delta}\right)\left(1-e^{-\mu_B\Delta}\right)}{\left(2-e^{-\lambda_A\Delta}-e^{-\mu_A\Delta}\right)\left(2-e^{-\lambda_B\Delta}-e^{-\mu_B\Delta}\right)}
\end{aligned}$$

$$\pi(11) = \frac{\left(1-e^{-\lambda_A\Delta}\right)\left(1-e^{-\lambda_B\Delta}\right)}{\left(2-e^{-\lambda_A\Delta}-e^{-\mu_A\Delta}\right)\left(2-e^{-\lambda_B\Delta}-e^{-\mu_B\Delta}\right)}$$

In scenario b), Alice and Bob have a communication relationship with each other and therefore, their behavior is not independent of each other, specifically when they are on a call with each other. Let (λ_{AB}, μ_{AB}) be the call arrival-service parameters for communication between Alice and Bob. Alice and Bob can jointly be in 5 possible states: (Alice is idle, Bob is idle), (Alice is idle, Bob is communicating with F_B), (Alice is communicating with F_A, Bob is idle), (Alice is communicating with F_A, Bob is communicating with F_B), and (Alice communicates with Bob) which are represented as (0,0), (0,1), (1,0), (1,1), and $(\overline{1,1})$ respectively. When Alice and Bob speak to each other, they must return to the idle state before establishing any other call as "Δ" is assumed to be sufficiently small to prohibit multiple transitions in one time-slot. In the case of a collision between calls from different parties arriving in the same time-slot, preference is given to calls between Alice and Bob; i.e., they can establish calls to third parties in a time-slot only if they do not receive a call from each other. This assumption is required to ensure that sum of transition probabilities equals 1 but does not particularly impact or favor our analysis. This is the case because Δ is sufficiently small such that the likelihood of independent call-arrivals from different parties to the same user in the same time-slot is small. The analysis can be done under an alternate assumption; e.g., priority to calls from third-parties, with minor modifications to the state transition probabilities. The collective state transition diagram for Alice and Bob's activity when they have a communication relationship is shown in Figure 2.4.

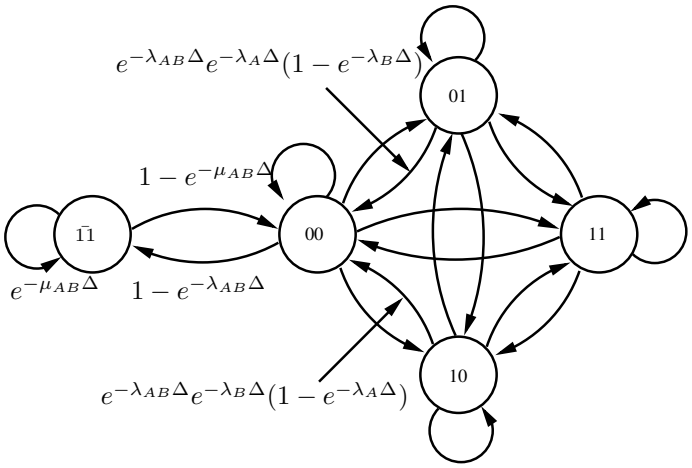


Figure 2.4: Joint transition diagram when a relationship exists between Alice and Bob

The stationary distribution $(\overline{\pi}_i)$ of this Markov chain can be calculated as:

$$\overline{\pi}_{00} = \frac{1}{1+\sum\limits_{i\in\{01,10,11,\overline{11}\}}\frac{\overline{\pi}_i}{\overline{\pi}_{00}}}$$

where,

$$\overline{\pi}_{01} = \frac{e^{-\lambda_{AB}\Delta}\left(1-e^{-\mu_A\Delta}\right)\left(1-e^{-\lambda_B\Delta}\right)}{\left(1-e^{-\mu_A\Delta}\right)\left(1-e^{-\mu_B\Delta}\right)}\overline{\pi}_{00}$$

$$\overline{\pi}_{10} = \frac{e^{-\lambda_{AB}\Delta}\left(1-e^{-\lambda_A\Delta}\right)\left(1-e^{-\mu_B\Delta}\right)}{\left(1-e^{-\mu_A\Delta}\right)\left(1-e^{-\mu_B\Delta}\right)}\overline{\pi}_{00}$$

$$\overline{\pi}_{11} = \frac{e^{-\lambda_{AB}\Delta}\left(1-e^{-\lambda_A\Delta}\right)\left(1-e^{-\lambda_B\Delta}\right)}{\left(1-e^{-\mu_A\Delta}\right)\left(1-e^{-\mu_B\Delta}\right)}\overline{\pi}_{00}$$

$$\overline{\pi}_{\overline{11}} = \frac{1-e^{-\lambda_{AB}\Delta}}{1-e^{-\mu_{AB}\Delta}}\overline{\pi}_{00}$$

Finally, Eve does not get to observe the actual calling behavior but only the activities of both the targets. In that case, it cannot distinguish between the states (1,1) and $(\overline{1,1})$. We denote "busy" as 1 and "idle" as 0 to represent the activity of a device, which makes the mapping from the calling status to activity-logs as shown in Figure 2.5.

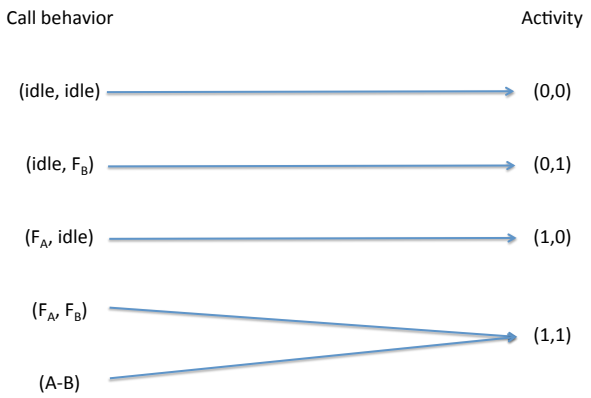


Figure 2.5: Mapping from calling behavior to activities

With the network and call arrival-service model specified, we move to the modeling of the busy/idle probing process of the attacker, in the next section.

2.2 Probing

The core of private communication detection is the detection of the busy/idle activity status of a target device. Side channel attacks can be employed to achieve this goal. In session initiation protocol (SIP) based VoIP networks, a resource saturation side channel can be exploited by an ordinary user of the network; i.e. without any special privileges, to acquire busy/idle activity status information for other users [35]. Every SIP device maintains a finite buffer that is used to store the context of protocol negotiation until a response is generated for the corresponding request. Each SIP request occupies one slot on the buffer. If the buffer gets full, the full-buffer-condition can be learned by the attacker due to generation of a different response by the target device. To perform busy/idle status detection, the attacker sends periodic SIP requests or probes (modified so that they don't alarm the target by ringing the device) to the target device and counts the number of probes required to cause the full-buffer-condition. Depending on the number of probes and the size of the buffer (as per the device specification), the attacker can learn if the device has an existing SIP request or not, thus revealing the busy/idle status of the device. Similarly, in Wi-Fi networks, the attacker can perform busy/idle status detection either by computing delay of his probes, as VoIP packets get priority, or sensing the Wi-Fi channel [33, 34].

Periodic probing strategy: To accurately model the periodic probing process of real-world attacks demonstrated by Jong and Gligor [34, 35], we assume that Eve sends probes to both Alice and Bob every T seconds and obtains their busy/idle status information over n samples. The time-gap between consecutive probes, T, is restricted to being an integer multiple of Δ; i.e. $T = r\Delta$ for some integer r. The probe rate of the attacker is defined as $1/T$. If $r > 1$, then the attacker only gets to observe the activity of Alice and Bob every r transitions. If M represents the probability transition matrix of calling behavior over consecutive time-slots, then the probability transition matrix over every r time-slots can be computed as M^r. As $r \to \infty$, M^r tends to the stationary distribution of the Markov chain. This implies that for small probe rate of the attacker, the observed activity at a given probe instance is independent of the activity observed in the past.

Observation noise: In the case of SIP-based VoIP networks, busy/idle status detection is done by observing the full-buffer-condition. This technique has no significant source of noise except packet drop which can be ignored for a wired medium. Therefore, we assume that the probing process of the attacker is noiseless. This assumption may not be true for other busy/idle detection mechanism, such as in Wi-Fi networks. However, modeling of noise in the attacker's observation process can always be done separately as a noisy channel between the true activity status and the activity status observed by the attacker.

Probe timeliness: Another assumption of this analysis is the timeliness of the probes, i.e. the probes do not incur significant propagation delay and notify the attacker about the instantaneous activity of the target. In the case of SIP-based VoIP networks, where packets are required to have transmission delay of less than 400 ms, this assumption is reasonable because the scale of call arrival-service is typically much larger than 400 ms. For example, a typical conversation lasts at least 30 seconds. Similarly, in the case of Wi-Fi networks where busy/idle detection can be done by sensing the channel, the probing delay is small to not cause any significant shift in the time-series of the activities of the two targets. Therefore, this assumption is justified.

Table 2.1: Notation

P^r	Probability transition matrix over joint calling behavior in the case of no communication relationship and probe rate $1/r\Delta$
$p^r_{i\to j}$	Individual transition probability from state i to j under P^r
π_i	Steady state probability of state i under P
$p^r(cr^n)$	Probability of observing a given call record cr^n under P^r
$\overline{P}^r$	Probability transition matrix over joint calling behavior in the case of communication relationship and probe rate $1/r\Delta$
$\overline{p}^r_{i\to j}$	Transition probability from state i to j under $\overline{P}^r$
$\overline{\pi}_i$	Steady state probability of state i under $\overline{P}$
$\overline{p}^r(cr^n)$	Probability of observing a given call-record (cr^n) under $\overline{P}^r$
$\mathcal{T}(al^n)$	Set of all call-record (cr^n) that map to al^n

Table 2.1 lists the notation used in this analysis. With the system model and assumptions in place, we analyze the accuracy of the attacker in detecting communication relationships in the next section.

2.3 Estimation of Communication Relationships

With the information gathered through a PCD attack on Alice and Bob, the first analysis that the attacker Eve can perform is to learn whether Alice and Bob have a communication relationship or not. This analysis can be formulated as a binary hypothesis testing problem where the null hypothesis (H_0) assumes that there is no relationship and the alternate hypothesis (H_1) assumes that a communication relationship exists. The Markov models shown in Figure 2.3 and 2.4 can be used to describe calling behavior of Alice and Bob under H_0 and H_1 respectively.

2.3.1 Analysis of the Maximum A-posteriori Probability (MAP) Detector

In the absence of gathered information; i.e. a-priori, Eve may have unequal biases towards H_0 and H_1. For example, in social scenarios, a suspicious person may have a strong reason to believe in the existence of the relationship between his/her partner and another person, which leads to him/her using PCD. In law-enforcement scenarios, intelligence received from other sources may indicate the existence of a relationship between Alice and Bob which requires confirmation by PCD. We denote, the a-priori probability of H_0 and H_1 as η and $1-\eta$ respectively, where $0 < \eta < 1$. After observing a n-length activity-log sequence (al^n) for Alice and Bob, the attacker can choose any detection rule $\mathcal{D}$ to detect the underlying hypothesis.

Definition 4. *The probability of error P_e achieved by a detection rule, $\mathcal{D}$, is defined as the probability that the attacker's estimate is incorrect.*

$$\begin{aligned} P_e &= P(\mathcal{D}(al^n) \neq \textit{the true hypothesis}) \\ &= P(H_0)P(\mathcal{D}(al^n) \neq H_0|H_0) + P(H_1)P(\mathcal{D}(al^n) \neq H_1|H_1) \end{aligned}$$

The choice of $\mathcal{D}$ to minimize P_e is the *maximum a-posteriori probability* (MAP) rule, which is specified as

$$MAP(al^n) = \begin{cases} H_0 & \text{if} (1-\eta)P(al^n|H_0) \geq \eta P(al^n|H_1); \\ H_1 & \text{if} (1-\eta)P(al^n|H_0) < \eta P(al^n|H_1) \end{cases}$$

The P_e achieved by the MAP rule can be bounded on the above, in terms of the system parameters, as follows:

$$\begin{aligned} P_e &= (1-\eta)\sum_{al^n} \min\left(P(al^n|H_0); \frac{\eta}{1-\eta}P(al^n|H_1)\right) \\ &\overset{a}{\leq} \sqrt{\eta(1-\eta)}\sum_{al^n}\sqrt{P(al^n|H_0).P(al^n|H_1)} \\ &\leq \sqrt{\eta(1-\eta)}\sum_{al^n}\sqrt{p^r(al^n).\sum_{cr^n \in \mathcal{T}(al^n)} \overline{p}^r(cr^n)} \end{aligned}$$

$$\begin{aligned} &\overset{b}{\leq} \sqrt{\eta(1-\eta)} \sum_{al^n} \left\{ \sum_{cr^n \in \mathcal{T}(al^n)} \sqrt{p^r(al^n).\overline{p}^r(cr^n)} \right\} \\ &\overset{c}{=} \sqrt{\eta(1-\eta)} ||\mathbf{V}\mathbf{X}^{(r)^n}||_1 \end{aligned} \tag{2.1}$$

where $||\ ||_1$ is the l_1 vector norm,

$$\mathbf{V} = [\sqrt{\pi_{00}\overline{\pi}_{00}}\sqrt{\pi_{01}\overline{\pi}_{01}}\sqrt{\pi_{10}\overline{\pi}_{10}}\sqrt{\pi_{11}\overline{\pi}_{11}}\sqrt{\pi_{\overline{11}}\overline{\pi}_{\overline{11}}}]$$

and $\mathbf{X}^{(r)}$ is a 5×5 matrix $\left[x_{i,j}^{(r)}\right]$ such that,

$$x_{i,j}^{(r)} = \begin{cases} \sqrt{p_{i\to j}^r \overline{p}_{i\to j}^r} & \text{for } i,j \in \{00,01,10,11\}; \\ \sqrt{p_{i\to 4}^r \overline{p}_{i\to 5}^r} & \text{for } i \in \{00,01,10,11\}, j = \overline{11}; \\ \sqrt{p_{4\to j}^r \overline{p}_{5\to j}^r} & \text{for } i = \overline{11}, j \in \{00,01,10,11\}; \\ \sqrt{p_{4\to 4}^r \overline{p}_{5\to 5}^r} & \text{for } i = j = \overline{11}; \end{cases}$$

a) Application of Bhattacharya bound; i.e. $\min(a,b) \leq \sqrt{ab}$, for $a, b > 0$.
b) For $a, b > 0$, $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$.
c) Representation of sum in terms of matrix multiplication.

Equation (2.1) shows that the upper-bound on P_e for the MAP rule is a concave function of η and is maximized for $\eta = 1/2$. This could lead to an incorrect assertion that the efficacy of PCD increases with increased bias of the attacker. In the case of PCD, another equally-important performance metric to consider is the probability of false positives; i.e. probability of incorrectly estimating communication between Alice and Bob even when there is none. In particular, if communication relationships revealed by PCD analysis are used as judicial evidence, then a threshold performance with regards to $P(\text{false} - \text{positives})$ may be mandated by law to prevent false indictments. We provide an upper-bound on $P(\text{false} - \text{positives})$ achieved by the MAP detector in terms of the communication and probing parameters, and the number of collected samples, n.

Theorem 2. *The probability of false positives,* $P(\text{false} - \text{positives}) \equiv P(MAP(AL^n) = H_1|H_0)$*, of a MAP detector is upper-bounded by:*

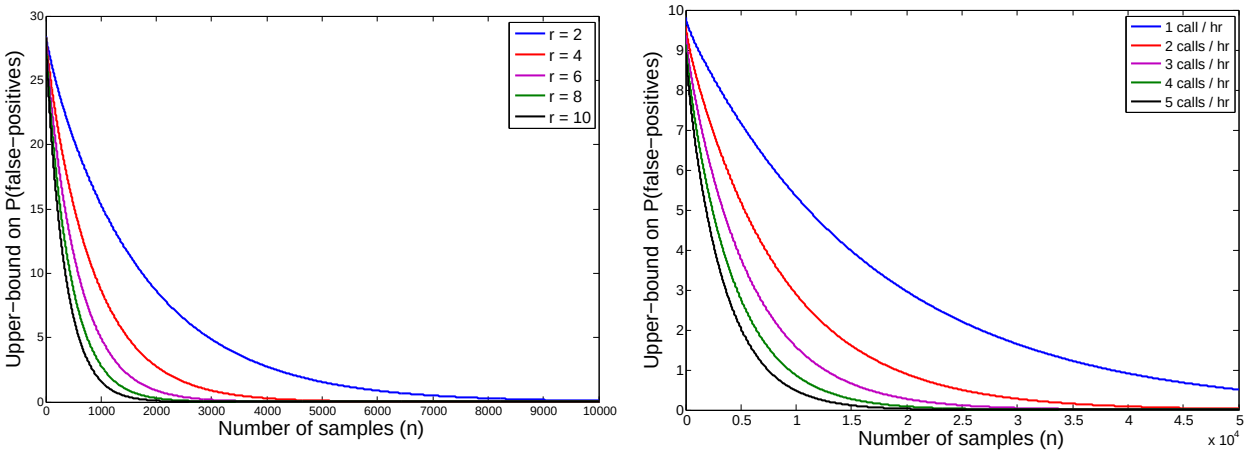
$$P(\text{false} - \text{positives}) \leq \sqrt{\frac{\eta}{1-\eta}} ||\mathbf{V}\mathbf{X}^{(r)^n}||_1$$

Proof. $P(\text{false} - \text{positives})$ is equal to the probability of the algorithm estimating H_1 whereas H_0 is the true hypothesis.

$$\begin{aligned} P_e &= P(H_0)P(MAP(al^n) = H_1|H_0) + P(H_1)P(MAP(al^n) = H_0|H_1) \\ &\geq (1-\eta)P(MAP(al^n) = H_1|H_0) \end{aligned}$$

This implies $P(\text{false} - \text{positives} \leq P_e/(1-\eta))$. From equation (1), we get the final inequality. □

Theorem 2 implies that to the attacker needs to collect more samples to achieve the same upper-bound on $P(\text{false} - \text{positives})$ as his a-priori bias towards H_1 increases. This ensures



(a) Effect of varying probe-gap, r

(b) Effect of varying call-rate, λ_{AB}

Figure 2.6: Upper bound on $P(\text{false} - \text{positives})$ vs the number of samples (n)

that the attacker bias does not dominate observed data, and acts as a safeguard. It is also important to understand the impact of other parameters, particularly the time gap between the attacker's probes, r, and the call arrival rate between Alice and Bob, λ_{AB}, on $P(\text{false} - \text{positives})$. Figure 2.6a shows that the upper-bound on the probability of false-positives decreases with increasing r. This is because for the same number of probes, n, a lower probe rate regime (higher r) observes the system for a longer time and is more likely to capture a communication between Alice and Bob. However, if the time-frame for the attack is fixed, a higher probe rate regime acquires higher number of samples and achieves higher accuracy. The plot of the accuracy of PCD for that case is not shown here. Figure 2.6b shows the plot of the probability of false-negatives vs λ_{AB}. Here, for a constant probe rate, the accuracy of PCD increases with λ_{AB} as the attacker is more likely to capture a conversation between Alice and Bob. Figures 2.6a and 2.6b were generated with the following parameter values: $\Delta = 0.5$ seconds, $\mu_{AB} = \mu_A = \mu_B = 300$ seconds, $\lambda_A = \lambda_B = 1$ call/hr, and $\eta = 0.0001$.

2.3.2 Analysis of the Neyman-Pearson Detector

While the MAP rule minimizes the average probability of error (P_e), in certain scenarios, it is relevant to minimize $P(\text{false} - \text{positives})$ and $P(\text{false} - \text{negatives})$ individually. As stated earlier, if communication relationships revealed by PCD are to be used as judicial evidence, then law may mandate a upper-limit, $\alpha \in (0, 1)$ on $P(\text{false} - \text{positives})$. The attacker, which may be a law-enforcement agency, may wish to minimize the $P(\text{false} - \text{negatives})$, subject to an upper-limit on $P(\text{false} - \text{positives})$. The decision rule that achieves this goal is the logarithmic ratio test (LRT), also known as the Neyman-Pearson detector [9], which is defined as

$$LRT(al^n) = \begin{cases} H_0 & \text{if} \log_2 \frac{P(al^n|H_1)}{P(al^n|H_0)} \geq \gamma; \\ H_1 & \text{if} \log_2 \frac{P(al^n|H_1)}{P(al^n|H_0)} < \gamma \end{cases}$$

γ is selected so that $P\left[\log_2 \frac{P(al^n|H_1)}{P(al^n|H_0)} < \gamma | H_0\right] = \alpha$. As the number of samples grows, $P(\text{false} - \text{negatives})$ reduces exponentially and the performance of the detector is measured in terms of the reliability

rate, which is defined as:

$$\text{reliability rate} = \lim_{n\to\infty} -\frac{1}{n}\log_2 P(\text{false}-\text{negatives})$$

For the LRT (or Neyman-Pearson detector), reliability rate can be computed as the Kullback-Leibler divergence rate [9],

$$\text{reliablity}-\text{rate} = \lim_{n\to\infty}\frac{D(P(AL^n)||\bar{P}(AL^n))}{n}.$$

Theorem 3. *The reliability rate, measured as the error-exponent of* $P(\text{false}-\text{negatives})$*, achieved by the Neyman-Pearson detector is computed as:*

$$\begin{aligned}\lim_{n\to\infty}\frac{D_{AL^n}(P||\bar{P})}{n} &= -\sum_{x,y}\pi_x\left[p^r_{x\to y}\log\left(\frac{\bar{p}^r_{x\to y}}{p^r_{x\to y}}\right)\right] \\ &\quad -\sum_{x,y}\pi_x\left[\sum_{k=1}^{\infty}p(x-(11)^k-y)\log\left(\frac{\bar{p}(x-(11/\overline{11})^k-y)}{p(x-11^k-y)}\right)\right]\end{aligned}$$

where, $x, y \in \{00, 01, 11\}$.

Proof. For concision, we present an intuitive argument for the proof. The probability of observing an activity-log (al^n) under H_0 is computed simply under the Markov model shown in Figure 2.3. Probability of al^n under H_1 is the sum of probabilities of all call records (cr^n) that map to al^n. However, the only sub-sequences of al^n that can be generated under different cr^n are of the type $x-(11)^k-y$, for $x, y \in \{00, 01, 10\}$, as these can be caused by call records sub-sequences of the type $x-(11/\overline{11})^k-y$. $\bar{p}(al^n)$ can be easily written in terms of the probabilities of these sub-sequences. Detailed steps are presented in the Appendix A.1. □

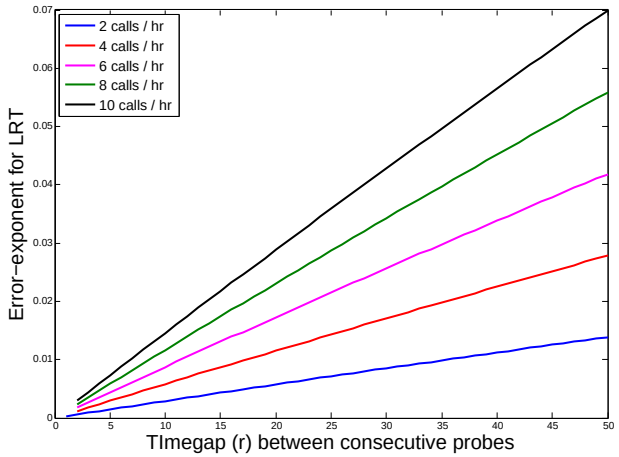


Figure 2.7: Reliability rate versus the probe time gap, r, plotted for different values of λ_{AB}

Figure 2.7 plots the reliability rate versus the probe time gap, r, for different values of calling rate between Alice and Bob, λ_{AB}. The plot was generated with the following parameter

values:$\Delta = 0.5$ seconds, $\mu_{AB} = \mu_A = \mu_B = 300$ seconds and $\lambda_A = \lambda_B = 1$ call/hr. Figure 2.7 clearly illustrates that the reliability rate increases with increasing calling rate between Alice and Bob, λ_{AB}. This is the case because higher calling rate implies greater separation between the Markov model for the two hypotheses. Similarly, the reliability rate increases with increasing probe time-gap, r, as the attacker observes the system for longer time periods.

In this section, we analyzed the accuracy of the attacker in detecting private communication between two targets and the effect of different parameters on it. We showed that the attacker's accuracy, measured in terms of average probability of error or probability of false-positives can be very high, demonstrating the threat of PCD. In the next section, we study the efficacy of PCD in revealing the private call records (length and time of calls) between Alice and Bob.

2.4 Estimation of Call Records

Once the attacker has been able to positively establish a communication relationship between Alice and Bob, he may wish to learn the call records of these targets, specially when they talk to each other. This knowledge allows the attacker to learn the frequency and pattern of the communication between Alice and Bob, providing further pertinent information about their identities and social relationships. In this section, we quantify the amount of information about call records leaked due to the activity logs collected by the attacker. On the one hand, this analysis can be used by the attacker to measure the reliability of the call records information he infers. On the other hand, this analysis enables a user, who cherishes his/her anonymity, to quantify the privacy provided by the communication system and compare it with other systems. To this end, we propose a metric to quantify the anonymity leakage ($\mathcal{L}$) of the system, based on the mutual information between the call records CR^n and the observed activity-logs AL^n.

Definition 5. *The anonymity leakage of a communication system for the two-target scenario is defined as:*

$$\mathcal{L} = \lim_{n\to\infty} \frac{I(CR^n; AL^n)}{H(CR^n)}$$

where $I()$ represents the mutual information and $H()$ represents the Shannon entropy *of the random variable concerned [11].*

The use of mutual information captures the reduction in uncertainty about the call records due to the knowledge of the activity-logs. At the same time, the mutual information is normalized with the a-priori entropy of the call records to provide a fair comparison between the different calling behaviors of the targets. This normalization also implies that $\mathcal{L} \in [0, 1]$. Finally, the limit is taken to observe the system in its steady state.

2.4.1 Anonymity Leakage with Noiseless Observations

We first compute the anonymity leakage when the attacker's busy/idle observation process is noiseless, as is the case with SIP-based VoIP networks. The leakage of the system under noiseless observation process also forms the worst-case scenario for the user's privacy and therefore, can be used to benchmark the weakness of the system. Any reduction in anonymity leakage due to a specific countermeasure can then be compared with the anonymity leakage under noiseless observations to measure the efficacy of the countermeasure. The underlying calling behavior

for the two-target scenario is as depicted in Figure 2.4.

Theorem 4. *The anonymity leakage of the system under noiseless observations is given by*

$$\mathcal{L} = 1 - \frac{\sum\limits_{x,y} \bar{\pi}_x \bar{p}^r_{x\to y} \log \bar{p}^r_{x\to y}}{-\sum\limits_i \overline{\pi}_i \sum\limits_j \overline{p}^r_{i\to j} \log \overline{p}^r_{i\to j}} - \frac{\sum\limits_{x,y} \sum\limits_{k=1}^{\infty} \bar{\pi}_x \overline{p}^r (x - (11/\overline{11})^k - y) \log \overline{p}^r (x - (11/\overline{11})^k - y)}{-\sum\limits_i \overline{\pi}_i \sum\limits_j \overline{p}^r_{i\to j} \log \overline{p}^r_{i\to j}}$$

where, $x, y \in \{00, 01, 11\}$.

Proof: The denominator $\lim_{n\to\infty} H(CR^n)/n$ can be simply computed as the entropy-rate of the first-order Markov chain. To compute the numerator, $I(AL^n; CR^n)$ can be written as $H(AL^n) - H(AL^n|CR^n)$. As the mapping from $cr^n \to al^n$ is a many-to-one mapping, the conditional entropy $H(AL^n|CR^n) = 0$. The entropy $H(AL^n) = -\sum \overline{p}(al^n) \log_2 \overline{p}(al^n)$. Detailed steps for computing $\overline{p}(al^n)$ are shown in the Appendix A.1.

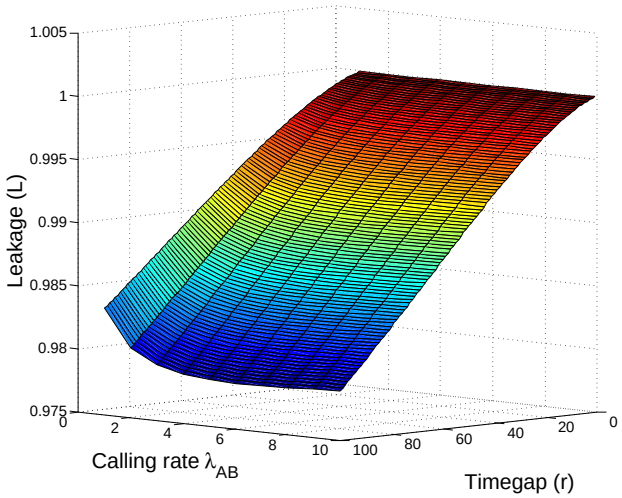


Figure 2.8: Plot of the leakage of the system vs the probe rate of the attacker, r, and calling rate between Alice and Bob, λ_{AB}

Figure 2.8 illustrates the plot of the leakage vs probe time gap, r and calling rate between Alice and Bob, λ_{AB}. The plot is generated under the following parameter values, $\Delta = 1$ second, $\mu_{AB} = \mu_A = \mu_B = 300$ seconds, and $\lambda_A = \lambda_B = 2$ calls/hr. The following inferences can be drawn from Figure 2.8

- The leakage of the system is very high, ≈ 1, which means that the attacker can get significant information about the call records by analyzing activity-logs.
- The system's leakage is high when call-arrival rate between Alice and Bob, λ_{AB}, is either lower or much higher than with their third-party friends. For preserving privacy against PCD attackers, users should homogenize their calling parameters among different parties so that communication with a particular party cannot be easily identified. This finding is validated by the real-world approaches employed by criminals in which they make spurious calls to other parties, in order to hide communication with a crime partner.

- The leakage of the system decreases as the probe time gap increases. However, even for a very large probe time gap, the system's anonymity leakage does not fall below 95%. This implies that PCD cannot be countered by restricting the attacker's probe rate.

2.4.2 Anonymity Leakage with Noisy Observations

The analysis so far assumed that the attacker's observation process is noiseless. Now, we study the impact of observation noise on the system's anonymity leakage. Such noise may be inherently present in the busy/idle observation process; e.g., in Wi-Fi networks, or may be intentionally introduced by countermeasures employed by the target device. We characterize this noise in the form of a communication channel $\mathcal{C}$ between the true busy/idle activity-log (al^n) and the observed busy/idle activity-log ($\hat{al}^n$), specified by the probability distribution $p(\hat{al}^n|al^n)$. If the channel $\mathcal{C}$ has a capacity $cap(\mathcal{C})$, then due to the definition of capacity [11]

$$I(AL^n; \hat{AL}^n) \leq \max_{p(al^n)} I(AL^n; \hat{AL}^n) = n \times cap(\mathcal{C}).$$

The system anonymity leakage $\mathcal{L}_{\mathcal{C}}$ under a noisy observation channel C is defined as,

$$\mathcal{L}_{\mathcal{C}} = \lim_{n\to\infty} \frac{I(CR^n; \hat{AL}^n)}{H(CR^n)}$$

As $CR^n \rightarrow AL^n \rightarrow \hat{AL}^n$ form a Markov chain, we have $I(CR^n; AL^n) \geq I(CR^n; \hat{AL}^n)$ or $\mathcal{L} \geq \mathcal{L}_{\mathcal{C}}$. This corresponds with intuition that observation noise reduces anonymity leakage and enhances the user's anonymity.

Definition 6. *The anonymity gain $\mathcal{G}_{\mathcal{C}}$ due to the presence of a noisy channel $\mathcal{C}$ between the real and observed activity-logs is given by:*

$$\mathcal{G}_{\mathcal{C}} = \frac{\mathcal{L} - \mathcal{L}_{\mathcal{C}}}{\mathcal{L}}$$

We immediately have,

Theorem 5. *For any channel $\mathcal{C}$ with channel capacity $cap(\mathcal{C})$*

$$\mathcal{G}_{\mathcal{C}} \geq 1 - \frac{cap(\mathcal{C})}{\mathcal{L} \times H_R(CR)}$$

where $\mathcal{L}$ is the leakage of the system with noiseless observations and

$$H_R(CR) = \sum_i \overline{\pi}_i \sum_j \overline{p}^r_{i\rightarrow j} \log \frac{1}{\overline{p}^r_{i\rightarrow j}}$$

is the entropy rate of the call records.

Proof: As $CR^n \rightarrow AL^n \rightarrow \hat{AL}^n$ form a Markov chain:

$$\begin{aligned} \lim_{n\to\infty} \frac{I(CR^n; \hat{AL}^n)}{n} &\leq \lim_{n\to\infty} \frac{I(AL^n; \hat{AL}^n)}{n} \\ &\leq cap(\mathcal{C}) \end{aligned}$$

Theorem 5 relates the anonymity gain of a noisy channel to its channel capacity. Lower capacity of a channel is a sign of more noise and leads to higher reduction in the leakage of the system.

So far, we have demonstrated the fact that PCD poses a major threat to user's anonymity. It can be used to detect private communication between users with high accuracy as well as to reliably learn their call records, even with low probe-rate of the attacker. Additionally, the capabilities required of an attacker to perform PCD are moderate: the attacker has to be a regular user of the communication network and has to know the contact information of the targets. This implies that unlike attacks which require substantial resources only governments or large organizations can provide, PCD can be launched by rogue individuals, putting anonymity of citizens at risk. It is vital to develop countermeasures that thwart PCD and protect user's privacy, as well as to quantify the efficacy of these countermeasure. In the next section, we analyze the same.

2.5 Countermeasures and their Analysis

There are a number of existing countermeasures to traffic-analysis attacks, such as firewalls, anonymous and virtual private networking. First, we analyze the efficacy of these countermeasures in preventing PCD and show that these countermeasures enjoy little or limited success in thwarting PCD.

2.5.1 Performance of Existing Traffic-analysis Countermeasures against PCD

Firewalls: Firewalls are typically used to block unwanted or suspicious packets/traffic patterns. As PCD works by sending periodic probes to the target device, blocking probe packets or probe traffic streams with the use of firewalls was proposed as a possible countermeasure against PCD [35]. However, setting up of firewall rules to block certain packet types is difficult as probe packets are indistinguishable from normal control packets. Under a different approach, firewalls can be used to block traffic patterns or rates that resemble the probe traffic. As the attacker sends periodic probes, the simplest approach is to block any traffic burst above a certain threshold, R_t. R_t must be suitably chosen to support legitimate voice packet streams and allow the normal functioning of the VoIP device. CODECS used for VoIP calling, such as the G.711, use a standard transmission rate of 50 packets/sec and therefore, it serves as a possible candidate for R_t. In the case SIP-based VoIP, different VoIP devices use different protocol buffer sizes B: $B = 32$ for Linksys PAP2 and 8 for Cisco 7490G [35] and therefore, the highest rate at which the attacker can probe the device is $50/B$ times per second. This limit on the maximum probe rate also enforces a limit on the maximum leakage of the system. Regrettably, even at such low probe rates the leakage of the system can still be significantly high, as shown in Figure 2.8. This, in effect, shows that the approach of limiting the attacker's probe rate does not hamper PCD but only reduces the accuracy of call records estimation marginally.

Virtual private networks (VPNs): VPNs allow users to securely access private networks from outside the network. However, they fail to prohibit PCD for two reasons. One, a VPN user is vulnerable to PCD being performed by another user within the VPN. In fact, the primary feature of PCD is that private call records information can be acquired by a network peer

without the requiring special network privileges. As specified earlier, the PCD attacker in this analysis is a user of the network. Second, even if the attacker is outside the VPN, he can perform PCD as long he can send probes and receive responses from the target device.

Anonymous networking: While low-latency anonymous networks protect information leakage through wiretapping and analysis of packet length/timing characteristics, they fail to prevent PCD as it exploits weaknesses present at the communication end-devices. Furthermore, the low-latency requirement of VoIP traffic ensures that irrespective of countermeasures deployed inside the network, the activity behavior of communicating end-devices is synchronous, enabling PCD.

As existing countermeasures to traffic-analysis are not effective against PCD, we develop a new countermeasure technique *resource-randomization* and prove its security. This technique is motivated by the analysis in Section 6.2 which shows that noise in the attacker's observation process reduces system anonymity leakage, decreasing the reliability of the information inferred by the attacker. Resource-randomization operates by randomizing the resource that is used for side channel observations, such as SIP buffer-size or packet delay. We develop and analyze countermeasures based on this technique for SIP-based VoIP networks and Wi-Fi networks.

2.5.2 Resource-randomization in SIP-based VoIP Networks

In the case of SIP-based VoIP networks, the busy/idle status detection is done by forcibly overflowing the SIP protocol buffer of the target device. Due to the use of a fixed-size buffer, the number of probes required to cause the full-buffer-condition (and receive an error message) has one-to-one correspondence with the activity status of the device which can then be estimated without error.

However, if the device is designed to randomly change the used buffer size at every time-slot, then the attacker will not be able to infer the underlying activity with the same accuracy. To highlight this, we start with a simple example. Let, the used buffer size B_u in a given time-slot be a random variable that takes the value B with probability 0.5 and $B-1$ with probability 0.5. In this case, the following inferences can be made by the attacker:

- Error after $B+1$ probes: The device is idle and $B_u = B$.
- Error after $B-1$ probes: The device is busy and $B_u = B-1$.
- Error after B probes: Either the device is idle and $B_u = B-1$, or the device is busy and $B_u = B$.

While the attacker can still make the correct inference when the error message is received after $B+1$ or $B-1$ probes, it cannot infer the activity status of the target correctly when the error message is received after B probes. Thus, this strategy creates a noisy channel between the activity status of the device and the number of probes required for buffer overflow, reducing leakage. From the analysis in the previous section, we know that the capacity of this noisy channel puts a limit on the leakage of the system. We compute the capacity of the channel by considering that the device is busy with probability p and idle with probability $1-p$. The probability p can be interpreted as the stationary probability of the target being idle/busy under

the Markov model of Figure 2.4. However, computation of channel capacity is done over all possible values of p. We have:

$$I(AL; \hat{AL}) = H(p/2, 1/2, (1-p)/2) - 1$$

The capacity of the channel is $\max_p I(AL; \hat{AL}) = 0.5$, achieved for $p = 0.5$. This example illustrates that even with two possible choices for the used buffer size, the system anonymity leakage be reduced substantially.

The technique explained above can be extended to achieve further anonymity gains by choosing B_u randomly from a larger set of values $\{B_{min}, B_{min}+1, \cdots, B_{max}\}$ s.t. $B_{max} - B_{min} = q$ with the uniform distribution. The noisy channel between the real activity status of the target and number of probes required to cause an error message is as shown in Figure 2.9. We again calculate the capacity of the channel in order to limit the system's leakage:

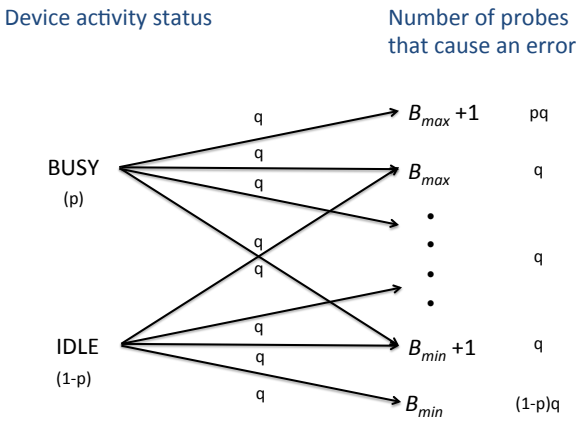


Figure 2.9: A visualization of the buffer randomization channel in SIP-based VoIP devices

Theorem 6. *The* capacity *of the buffer randomization channel visualized in Figure 2.9 is given by:*

$$cap(\text{buffer} - \text{randomization}) = \frac{1}{B_{max} - B_{min} + 1}$$

Proof: Simple calculation of the capacity of a discrete memoryless channel.

Theorem 6 proves that system designers can form a channel with an arbitrarily low capacity by simply increasing the size of the set from which the used buffer size can be chosen. From Theorem 5, as the capacity of the channel goes to 0, the anonymity gain $\mathcal{G}$ goes to 1, i.e. the system provides perfect anonymity against PCD attacks.

2.5.3 Resource-randomization in Wi-Fi Networks

In Wi-fi networks, the busy/idle status detection can done by sending periodic probe requests to the target and measuring delay in the probe responses. If a target is busy on a VoIP call, the probe requests will be served after the VoIP packets and therefore, the reply will take more

time compared to a device that is idle. In a practical setting, this difference in timing is variable due to the presence of network jitter and other delays in the network. We create an ideal attack scenario for the attacker by assuming that such variations are absent, and show that even in this case, randomization can be used to reduce the leakage of the system. Let T_1 be the time taken by the reply when the device is idle and $T_2 > T_1$ be the time taken by the reply when the device is busy. The attacker can identify the activity status of the device by simply checking the time taken by the reply, making the busy/idle status detection process noiseless.

To prohibit this, a random delay is added to the reply to introduce noise in the attacker's observation process. Let the random delay D be chosen uniformly from the range $(0, D_{max})$. Then, the time taken by the reply T is $T_1 + D$ or $T_2 + D$ depending on the state of the device. Figure 2.10 visualizes the noisy channel in the busy/idle detection process. Let p be the probability of the device being busy and $1 - p$ be the probability of the device being idle. The mutual information between the input and the output of the channel, i.e. the status of the device and the time taken by the reply respectively, equals $H(p)\left(\frac{T_2 - T_1}{D_{max}}\right)$. The maximum value of the mutual information or the channel capacity is achieved for $p = 1/2$. This result shows that the capacity of the channel can be made arbitrarily small by increasing the maximum delay D_{max} and therefore PCD can be completely thwarted by adding large random delay.

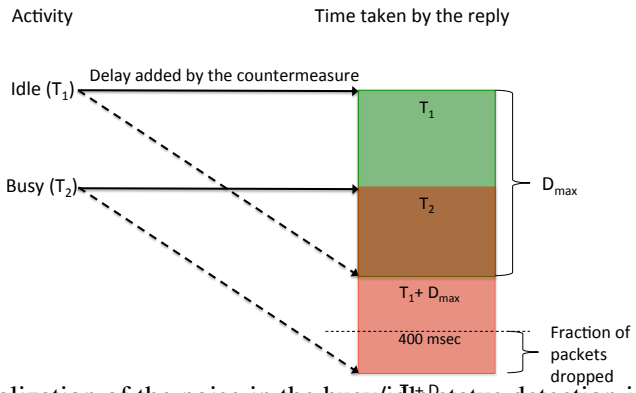


Figure 2.10: A visualization of the noise in the busy/idle status detection in Wi-Fi networks introduced by the countermeasure

However, in this case, resource randomization suffers from practical considerations. Addition of large delay might make packets useless for certain purposes. For example, the network delay of VoIP packets must be less than 400 ms as per the ITU-T recommendations. Addition of random delay by the countermeasure would lead to the total delay of packet crossing the 400 ms limit and being dropped. The choice of the maximum delay D_{max} must be made in such a way that the fraction of the packets dropped is within the acceptable threshold as per the system specification. For specified values of $T_1, T_2 < 400$ ms and D_{max}, the fraction of packets dropped due to the addition of the random delay by the countermeasure is given by $\frac{1}{2}\left(\frac{D_{max}+T_1-400}{D_{max}} + \frac{D_{max}+T_2-400}{400}\right)$. If $\theta > 0$ is the maximum fraction of packet drops for acceptable service, then the maximum permissible value of D_{max} is given by:

$$D_{max} = \frac{400 - \frac{T_1+T_2}{2}}{1-\theta}$$

This in turn, provides a lower bound on the capacity of the noisy channel or a cap on the efficacy of the countermeasure.

The analysis done in this section shows that resource-randomization can successfully thwart PCD. At the same time, its practical implementation in real-world communication devices is easy. Together, these features of resource-randomization have positive implications on the privacy of users against PCD. The success of this technique also implies that while designing practical communication devices, the amount of resources allocation to a particular task should not be deterministic to ensure that usage of these resources cannot be attributed to a specific behavior of the device, preventing side-channel attacks.

2.6 Conclusions

Private communication detection can be a powerful tool for governments, corporations, and rogue individuals that wish to extract information such as communication relationships of their targets and might be desirable for its low cost. As information extracted by this attack might be used in the future as actionable evidence for further privacy breach, it is important to understand the strengths and limitations of this attack, and provide performance guarantees. In this work, we have developed a quantitative framework to understand the impact of the probing strategy of the attacker and the calling behavior of the users on the efficacy of PCD in determining communication relationships and call records. We have developed mathematical guarantees on the efficacy of communication relationship classification and the leakage of the communication record information. At the same time, we have analyzed the efficacy of different countermeasures, such as resource randomization and firewalls, in thwarting PCD attacks under the same leakage model. Our results show that resource randomization outperforms firewall protection as it introduces noise in the side channel used to observe communication activity. This analysis provides a set of tools that can help system designers in building provably secure systems.

Chapter 3

Side Channels in Shared Network Components

In the presence of end-to-end encryption of packet contents and headers; e.g., in TLS and IPSec, the focus of security attacks is shifting towards traffic-analysis. Packet schedulers form an integral part of the Internet infrastructure and are used for tasks ranging from packet forwarding to traffic management. Due to the amount of traffic handled by them, packet schedulers have become valued targets for traffic-analysts. Even if the correlation of incoming and outgoing traffic at routers and forwarding node is removed with the use of anonymizing techniques, side-channel attacks can been used to de-anonymize traffic. Packet schedulers have finite resources so they queue packets belonging to different traffic streams and forward them using policies such as first-com-first-serve (FCFS). This limitation leads to dependence between the packet delay of one stream and the traffic generated by another stream. An attacker can use the delay information of his packets and estimate the traffic pattern of a private stream. Learned traffic patterns can be used to cause significant privacy and anonymity breaches. For example, Gong *et al.* used estimated traffic patterns of a user's traffic stream at a DSL router and compared them with traffic patterns of known websites to reveal the identity of the website [28]. Similarly, Murdoch and Danezis demonstrated an attack where the attacker can reveal the secret path used by an anonymous stream in Tor. This is achieved by transmitting a specific traffic pattern on the user's chosen OR path and using side channels to identify if sent traffic pattern is forwarded by a target OR. In both attacks, the attacker periodically probes the target device and correlates the delay of probe responses with known traffic patterns. Figure 3.1 illustrates the general attack setup.

A novel information-theoretic analysis of this side channel was performed by Gong *et al.* [27] for a two-user shared FCFS packet scheduler. In their analysis, the user's traffic stream and attacker's probe stream were modeled as Bernoulli-distributed random processes and the scheduler was assumed to serve one packet per time-slot. They measured the vulnerability of the scheduler in terms of the leakage of user's packet arrival pattern due to attacker's knowledge of his probes' delay. Their results showed that the leakage of the scheduler approaches maximum; i.e. 1, when the attacker probes at the fastest possible rate. While these attack examples and analysis highlighted the threat posed by this side channel, they were limited in the sense of considering the worst-case impact. The two attack demonstrations are performed in laboratory settings which were favorable to the demonstrations. For example, Murdoch and Danezis used a small Tor network that comprised of 13 Onion Routers and Gong *et al.* assumed that the user

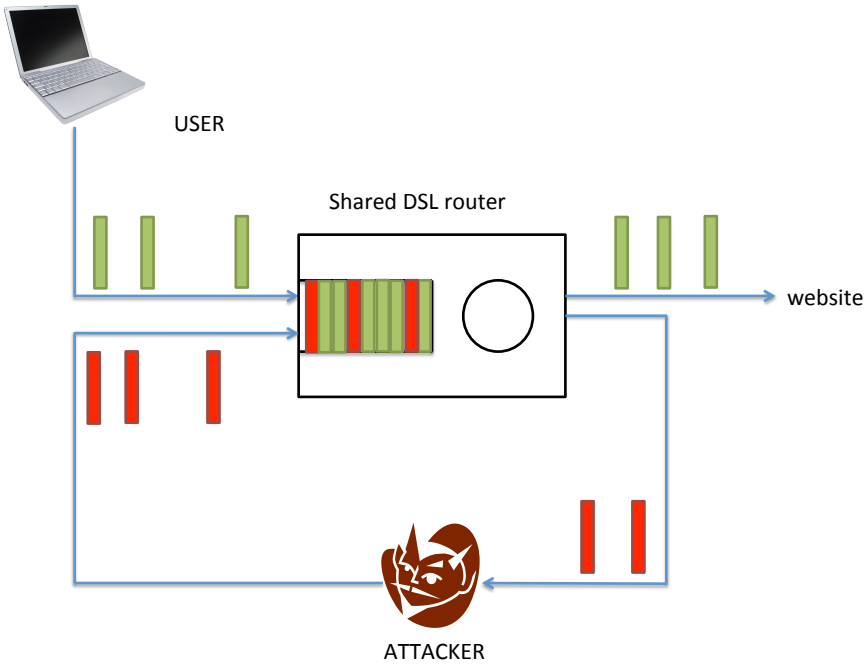


Figure 3.1: Privacy breach through side channel attack on a shared packet scheduler

only visits one website at a given time without any background traffic. These assumptions don't hold in practice; e.g., a real-world Tor network comprises of around 1300 ORs. To successfully launch these attacks in a real setting requires significant bandwidth resources from the attacker.

This limitation exists, mainly due to the inefficient use of bandwidth by the attacker in both cases. Periodic probing is a simplistic strategy but it requires the same probe rate even if the scheduler is already fully-clogged. Intuitively, a more intelligent and adaptive attacker will stop sending probe traffic in that circumstance and save up on bandwidth. Excessive and periodic probing as used in the literature is also more likely to be identified and blocked, defeating the purpose of the attack. A natural question that arises from this discussion is whether the attacker can find better, yet optimal, probing strategies for a given bandwidth budget and how much benefit can be extracted by such strategies. Clearly, this question is pertinent to other, possibly all, side-channel attack setups. In this chapter, we use the example of packet schedulers and build on the model developed by Gong *et al.* to answer both questions positively. Specifically,

- We show that a non-adaptive attacker; i.e. one that does not rely on previously collected information to decide future inputs, can achieve upto 1000% enhancement in information leakage for the same bandwidth budget as Geometric probing strategy used by Gong *et al.*
- We discuss the limitations of leakage metrics used in the literature and give rationale behind the use of causally-conditioned entropy in the leakage metric defined in Section 1.
- Under our metric, we show that an adaptive attacker; i.e. one that uses past observations to decide future inputs, can achieve a further 30% gain over optimal non-adaptive strategies.

- Finally, we find optimal real-world strategies that incorporate delay in feedback present in a real-world setup and compute their leakage.

In each case, we show that optimal strategies can be identified by solving linear programs that bring them under the reach of a computationally-constrained attacker. Such enhancements on leakage show that side-channel attacks on packet schedulers have the potential of breaking anonymity in real-world and large-scale networks. They also highlight the limitation of countermeasures, such as *privacy-preserving scheduling* [36], which only consider specific and non-optimal attack strategies. We now describe the theoretical model developed by Gong *et al.* [27] with modifications to generalize attack strategies.

3.1 System Description

The basic setup consists of a user and an attacker that share a FCFS packet scheduler. Time is discretized into fix-sized time-slots and packet arrivals occur only at the beginning of a time-slot. In case of a collision between the user's packet and attacker's packet arriving on the same time-slot, priority is given to the attacker's packet. The scheduler serves one packet per time-slot following FCFS scheduling and service for each packet take exactly one time-slot.

Let,

- t_i represent the time of arrival of the i^{th} probe to the queue,
- t'_i represent the time of departure of the i^{th} probe from the system,
- $a_i = t_i - t_{i-1}$ represents the inter-arrival time for the i^{th} probe,
- $d_i = t'_i - t_i$ represents the delay of the i^{th} probe including the fixed service time of one time-slot,
- x_i represent the number of user's packets arriving to the system in the time interval $[t_{i-1}, t_i)$,
- t^n, t'^n, a^n, d^n, x^n represent the collection of respective items for n probes. item Capital letters represent random variables and small letter their realization.

3.1.1 User's Packet Arrival Process

Packets arrive from the user following a Bernoulli arrival process with rate λ_1. That is, in each time-slot a maximum of one packet arrives from the user to the scheduler with probability λ_1 and no packet arrives with probability $1 - \lambda_1$. Packet arrivals in different time-slots are independent and identical random processes. Thus, given a specific time interval $[t_1, t_2)$, the number of packets arriving from the user $x \in \{0, 1, \cdots, t_2 - t_1\}$ with

$$p(x|t_2 - t_1) = \binom{t_2 - t_1}{x} \lambda_1^x (1 - \lambda_1)^{t_2 - t_1 - x}$$

3.1.2 Attacker's Probing Strategy

The attacker's probing strategy is characterized by his choice of the probability distribution on the inter-arrival times of his probes (a_i) for $a_i \in [1, 2, \cdots, \infty)$. The attacker may choose the inter-arrival time of the n^{th} probe, a_n, by considering all the previous observed inter-arrival times, a^{n-1} and queuing delays, d^{n-1} ; i.e. through a probability distribution $p(a_n|a^{n-1}, d^{n-1})$. This represents the most general form of feedback in the side channel. The attacking strategy is, however, limited by the average probe arrival rate (λ_2). The average probe arrival rate is computed as the inverse of the average probe inter-arrival time

$$\lambda_2 = \frac{1}{\sum_a ap(a)}$$

In case of non-adaptive attacks, the attacker may choose any time-invariant strategy $p(A_n = a) = p(a)$. The inter-arrival times for different packets are identical and independent random processes. Gong *et al.* analyzed the system leakage for an non-adaptive Bernoulli attack process; i.e.

$$p(A_n = a) = (1 - \lambda_2)^{a-1}\lambda_2$$

3.1.3 System Leakage

The leakage $\mathcal{L}$ of the system is defined as [27]:

$$\mathcal{L} = \lim_{n\to\infty} 1 - \frac{H(X^n|T^n, T'^n)}{H(X^n|T^n)}$$

where $H()$ represents the Shannon entropy of the input random variable [11].

3.1.4 Side-channel Model

The value of x_i depends only on the inter-arrival time for the i^{th} packet, the service delay of the $i - 1^{th}$ and the i^{th} packet; i.e. $a_i = t_i - t_{i-1}$, $d_{i-1} = t'_{i-1} - t_{i-1}$, and $d_i = t'_i - t_i$ respectively [27]. Due to this relationship, we can model the side channel as a two-user system shown in Figure 3.2. The input of the attacker to the side channel is a_i which is the inter-arrival

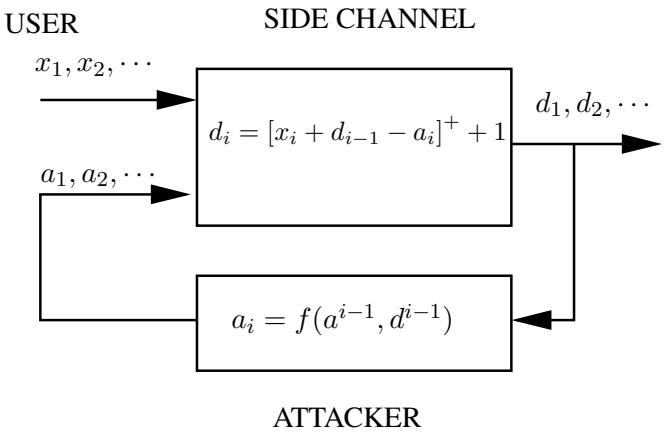


Figure 3.2: A model for the side channel at a FCFS packet scheduler

time for the i^{th} probe chosen according to a chosen distribution $p(a)$. The input of the user to the side channel is $x_i \in \{0, \cdots, a_i\}$ which is the number of user's packet that arrive to the scheduler in the period t_i generated following the Bernoulli process. The output of the side channel is d_i which is the delay for the i^{th} packet. This output is made available to the attacker who may or may not use it to generate the next input a_{i+1}. The delay-traffic side channel is then defined as the fundamental relationship between the inputs to the channel and the output:

$$d_i = [x_i + d_{i-1} - a_i]^+ + 1 \tag{3.1}$$

In the next section, we formulate the information leakage for all non-adaptive attacking strategies and identify the strategies that cause the maximum possible leakage.

3.2 Optimal Non-adaptive Strategies for Information Leakage

3.2.1 Leakage of General Non-adaptive Strategies

To identify the optimal non-adaptive strategy for a given probe rate, we first analyze the leakage of a general attack strategy. A non-adaptive attacker chooses a time-invariant attack strategy that picks the inter-arrival time for each probe independently of previous probes and side-channel observations. That is, the inter-arrival time of the n^{th} probe, $p(a_n = a) = p(a)$ for any n. The chosen distribution is subject to an average probe rate constraint of λ_2; i.e.

$$\sum_a ap(a) = \frac{1}{\lambda_2}$$

For such strategies, leakage can be computed as,

$$\begin{aligned}\lim_{n\to\infty} 1 - \frac{H(X^n|A^n, D^n)}{H(X^n|A^n)} &= 1 - \lim_{n\to\infty} \frac{H(X^n|A^n, D^n)}{H(X^n|A^n)} \\ &= 1 - \frac{\lim_{n\to\infty} \frac{H(X^n|A^n, D^n)}{n}}{\lim_{n\to\infty} \frac{H(X^n|A^n)}{n}}\end{aligned}$$

This implicitly assumes that the denominator is non-zero. The limits in the numerator and denominator can be further computed using Césaro's Mean Theorem [48], which states that if a sequence $\{z_i\}$ converges to z, then the running-average $\frac{\sum_{i=1}^{n} z_i}{n}$ also converges to z. We first compute the denominator, $H(X^n|A^n)$, as

$$\begin{aligned}H(X^n|A^n) &= \sum_{i=1}^{n} H(X_i|A_i) \\ &= n\sum_{a=1}^{\infty} p(a) H_B(\lambda_1, a)\end{aligned}$$

$$\lim_{n\to\infty} \frac{H(X^n|A^n)}{n} = \sum_{a=1}^{\infty} p(a) H_B(\lambda_1, a)$$

where, $H_B(\lambda_1, a)$ represents the entropy of the binomial distribution with success probability λ_1 and number of trials a. This is because the given the knowledge of the time-period, the number of user's packet arrival follow the binomial distribution. $H_B(\lambda_1, a)$ is easily computed as $\frac{1}{2}\log_2(2\pi e a \lambda_1(1-\lambda_1)) + \mathcal{O}(\frac{1}{a})$.

Similarly, for the numerator

$$\begin{aligned} H(X^n|A^n, D^n) &= H(X^n|A^n, D^n) \\ &= \sum_{i=1}^{n} H(X_i|X^{i-1}, A^n, D^n) \\ &= \sum_{i=1}^{n} H(X_i|A_i, D_i, D_{i-1}) \end{aligned}$$

This is because of the mathematical relationship between x_i, a_i, d_i, and d_{i-1} which derives from the basic distributions of the side channel (Equation 3.1). Moreover, if $d_i > 1$, x_i can be determined in terms of other parameters with certainty. That is, $H(X_i|a_i, d_i > 1, d_{i-1}) = 0$. This implies that the entropy of the attacker in estimating X_i is non-zero only if a probe arrives to experience an empty queue. If $d_i = 1$, then from Equation 3.1 we can see that $x_i \in \{0, \cdots, a_i - d_{i-1}\}$. $H(X_i|a_i, d_i, d_{i-1})$ can be computed as

$$\begin{aligned} H(X_i|A_i, D_i, D_{i-1}) &= \sum_{a_i, d_{i-1}} P(a_i)P(d_{i-1})P(d_i = 1|a_i, d_{i-1})H(X_i|a_i, d_i = 1, d_{i-1}) \\ &= \sum_{a_i, d_{i-1}} P(a_i)P(d_{i-1})P_{EQ}(a_i, d_{i-1})H(X_{a_i, d_{i-1}}) \end{aligned}$$

where,

- $P_{EQ}(a_i, d_{i-1})$ represents the probability of observing an empty queue given delay of the previous probe and the inter-arrival time of the current probe. $P_{EQ}(a_i, d_{i-1}) = P(X_i \leq a_i - d_{i-1})$
- $X_{a_i, d_{i-1}}$ is a random variable that represents the number of user's packets that arrive between two consecutive probes such that an empty queue can be caused. Then,

  $$P(X_{a_i, d_{i-1}} = x) = \frac{\binom{a_i}{x}\lambda_1^x(1-\lambda_1)^{a_i - x}}{\sum_{x=0}^{a_i - d_{i-1}} \binom{a_i}{x}\lambda_1^x(1-\lambda_1)^{a_i - x}}$$

  for $x \in \{0, 1, \cdots, a_i - d_{i-1}\}$.

Again, Césaro's mean theorem can be applied to compute

$$\lim_{n\to\infty} \frac{H(X^n|A^n, D^n)}{n} = \lim_{n\to\infty} H(X_n|A_n, D_n, D_{n-1})$$

provided the limit of each term in the expansion of $H(X_n|A_n, D_n, D_{n-1})$ exists. Since the probe distribution, $p(a)$, is time-invariant and both $P_{EQ}(a_n, d_{n-1})$, $H(X_{a_n, d_{n-1}})$ depend entirely

on the values of a_n and d_{n-1}, and are independent of the other parameters. Now, we show that the queue length behaves as a first-order, irreducible Markov chain and therefore, the limiting distribution $\lim_{n\to\infty} P(d_{n-1})$ exists and equals the stationary distribution of the Markov chain. Let, π_q denote the stationary probability of the queue length being q.

Theorem 7. *The stationary probability "π_q" of a probe experiencing a delay of $q \in \{1, \cdots \infty\}$ upon its arrival is given as $\pi_q = \alpha^{q-1}(1-\alpha)$, where α is the solution of the equation*

$$\alpha = \sum_{a=1}^{\infty} p(a)(\lambda_1 + \alpha(1-\lambda_1))^a$$

For binomial probing with rate parameter λ_2 used in [27],

$$\alpha = \frac{\lambda_1\lambda_2}{(1-\lambda_1)(1-\lambda_2)}.$$

Proof. Let the queuing delay experienced by the n^{th} probe $D_n = a$. Then, the queuing delay experienced by the $n+1^{th}$ probe, $D_{n+1} \in \{1, \cdots, a+1\}$. This is because either the inter-arrival time for the $n+1^{th}$ probe, a_{n+1} is large and the number of packets of the user x_{n+1} small enough for the queue to drain, or a packet arrives from the user in each time-slot: i.e. $x_{n+1} = a_{n+1}$ so that the queue length increases by one. Importantly, the queue length cannot increase more than one between two consecutive probes, and the queue length cannot reduce by more than a_{n+1} between consecutive probes. Define

$$\gamma_i := \sum_{a=i}^{\infty} p(a)\binom{a}{i}(1-\lambda_1)^i\lambda_1^{a-i}$$

and transition probability $P(D_{n+1} = b | D_n = a) := p_a^b$ Then,

$$p_a^b = \begin{cases} \gamma_{a-b+1} & \text{for } b > 1 \\ \sum_{i=a+1}^{\infty}\gamma_i & \text{for } b = 1 \end{cases}$$

As the state transition probability is independent of n and depends only on the previous state, this stochastic process is essentially a first-order stationary Markov process. Also, the transition probabilities imply that the the Markov chain is a-periodic and irreducible. Therefore, a unique stationary distribution for the Markov chain exists [30]. Let, $\pi_1, \pi_2, \cdots$ represent the stationary distribution of the process. Then, the global balance equations for the Markov chain are give as

$$\begin{bmatrix} p_1^1 & p_2^1 & p_3^1 & \cdots \\ p_1^2 & p_2^2 & p_3^2 & \cdots \\ 0 & p_2^3 & p_3^3 & \cdots \\ 0 & 0 & p_3^4 & \cdots \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \vdots \end{bmatrix}$$

Replacing p_a^b with γ_{a-b+1} for $b > 1$, we get

$$\begin{bmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \gamma_3 & \cdots \\ 0 & \gamma_0 & \gamma_1 & \gamma_2 & \cdots \\ 0 & 0 & \gamma_0 & \gamma_1 & \cdots \\ 0 & 0 & 0 & \gamma_0 & \cdots \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \vdots \end{bmatrix} = \begin{bmatrix} \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \\ \vdots \end{bmatrix}$$

For $\pi_i = \pi_1 \alpha^{i-1}$, all the above balance equations convert to a single balance equation

$$\gamma_0 + \alpha\gamma_1 + \alpha^2\gamma_2 + \cdots = \alpha$$

Additionally, due to the condition $\sum_i \pi_i = 1$, we have $\pi_1 = 1 - \alpha$ and $\pi_i = \alpha^{i-1}(1-\alpha)$, where $\alpha \in [0, 1]$ to maintain $\pi_i \in [0, 1]$. Rewriting the above equation, we get

$$\begin{aligned} \alpha &= \sum_{i=0}^{\infty} \gamma_i \alpha^i \\ &= \sum_{i=0}^{\infty} \left(\sum_{t=i}^{\infty} p(a) \binom{a}{i} (1-\lambda_1)^i \lambda_1^{a-i} \right) \alpha^i \\ &= \sum_{i=0}^{\infty} \left(\sum_{t=i}^{\infty} p(a) \binom{a}{i} (\alpha(1-\lambda_1))^i \lambda_1^{a-i} \right) \\ &= \sum_{a=1}^{\infty} p(a) \left(\sum_{i=0}^{a} \binom{a}{i} (\alpha(1-\lambda_1))^i \lambda_1^{a-i} \right) \\ &= \sum_{a=1}^{\infty} p(a)(\lambda_1 + \alpha(1-\lambda_1))^a \end{aligned}$$

Equivalently, $\alpha = \frac{\beta - \lambda_1}{1-\lambda_1}$, where β is the solution of the equation

$$\frac{\beta - \lambda_1}{1-\lambda_1} = G_p(\beta)$$

Here, $G_p()$ represents the standard probability generating function of the inter-arrival distribution. □

With this, we can simplify leakage for a general non-adaptive strategy, $p(a)$, as

$$\lim_{n\to\infty} 1 - \frac{H(X^n|A^n, D^n)}{H(X^n|A^n)} = 1 - \frac{\sum_{a,d} p(a)\pi(d)P_{EQ}(a,d)H(X_{a,d})}{\sum_a p(a)H_B(\lambda_1, a)}$$

3.2.2 Optimal Non-adaptive Strategies

To find the optimal non-adaptive strategy, specified as the inter-arrival distribution $p(a)$, we first fix an α which converts the balance equations for the stationary distribution into a linear

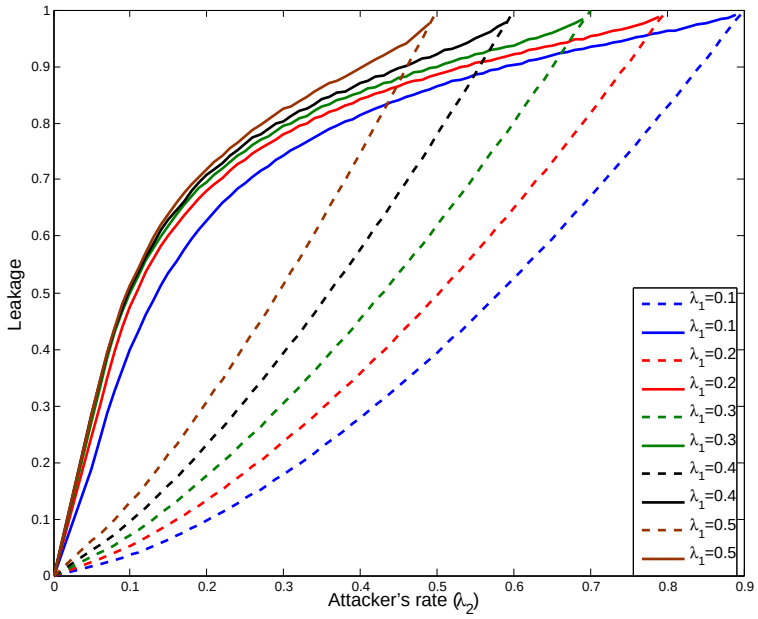


Figure 3.3: Leakage with optimized non-adaptive attacks vs binomial probing

constraint in terms of the control variables $p(a)$. Similarly, the average probe rate constraint $\sum ap(a) = 1/\lambda_2$ is another linear constraint in $p(a)$. For a given α, the optimal strategy is found as:

Maximize

$$\frac{\sum_{a,d} p(a)\left\{\pi(d)\left(H_B(\lambda_1, a) - P_{EQ}(a,d)H(X_{a,d})\right)\right\}}{\sum_{a} p(a)H_B(\lambda_1, a)}$$

Subject to

1: $0 \leq p(a) \leq 1$, for all a

2: $\sum_{a=1}^{\infty} p(a = 1) = 1$

3: $\sum_{a=1}^{\infty} ap(a) = 1/\lambda_2$

4: $\sum_{a=1}^{\infty} p(a)(\lambda_1 + (1-\lambda_1)\alpha^*)^a = \alpha^*$

where α^* is the maximum $\alpha \in (0,1)$ such that the above linear program has a solution. The reason to choose the largest α is because it ensures higher probability of a clogged queue and therefore, increases leakage.

Note that the objective function here is a linear-fractional function: i.e. the numerator and denominator are linear function of the control variables. Linear fractional functions are *quasi-linear* and therefore, have a unique maximum (or minimum value) which can be discovered by solving an equivalent linear program [5]. Simply, to maximize $\frac{\sum_i g_i x_i}{\sum_i h_i x_i}$ subject to linear constraints $\sum_i k_i x_i = 0$, one needs to find the largest t such that the linear system $\sum_i (g_i - t h_i) x_i = 0$ and $\sum_i k_i x_i = 0$ has a solution. This solution is the optimal strategy and corresponding leakage is the maximum leakage. As an example, for $\lambda_1 = 0.1$, $\lambda_2 = 0.1$, and $1 \leq a \leq 50$, the distribution that achieves the maximum leakage is $p(1) = 0.8075$, $p(2) = 0.0090$, and $p(50) = 0.1835$ and the corresponding leakage is 0.4. For the same λ_1, λ_2, geometric probing achieves a leakage of 0.0361. Figure 3.3 illustrates the leakage of the optimal strategies (solid lines) and geometric probing (dotted lines) versus the specified average probe arrival rate of the attacker (λ_2). The comparison is done for five different average packet rate of the user $\lambda_1 = \{0.1, 0.2, 0.3, 0.4, 0.5\}$. Clearly, the leakage under optimal non-adaptive attack strategies is significantly higher than for geometric probing for the same average probe rate.

Next, we show that an adaptive attacker; i.e. one that uses previous observations to determine future inputs, can potentially achieve even more leakage than optimal non-adaptive attacker. However, we first highlight some limitations of the leakage metric used by Gong *et al.* [27] in analyzing adaptive strategies. These limitations stem from the processing of causal information which, while not being important for non-adaptive strategies, leads to erroneous measurement of leakage for adaptive strategies. To recommend necessary changes in the leakage metric to overcome these issues.

3.3 *Causal Leakage*: a New Leakage Metric for Adaptive Attack Strategies

To drive the discussion on leakage metrics, we first discuss the intuition behind them. The leakage metric intends to capture the reduction in entropy of the user's input given the attacker's knowledge of side-channel outputs. Basically, the metric quantifies amount of additional information about user's inputs that is provided due to the attacker knowing side-channel outputs. Baseline comparison is performed with the a-priori information that the attacker has about the user's input when he only knows his own inputs. This is represented in the denominator of the leakage expression, $H(X^n|A^n)$. It is important to point out that the side channel is causal; i.e. an output depends only on past inputs and independent of future inputs to the side channel. Using the chain rule of entropy [11], we get

$$H(X^n|A^n) = \sum_i H(X_i|X^{i-1}, A^n)$$

For non-adaptive strategies, the denominator indeed captures the baseline (a-priori) information possessed by the attacker. As future inputs are independent of past inputs/outputs, X_i depends only on the past information; i.e. $\hat{A}_i$ which leads to the reduction $H(X^n|A^n) =$

$\sum_i H(X_i|A_i)$. Such independence does not exist for adaptive strategies because the knowledge of future side-channel inputs inadvertently implies knowledge of past outputs as the attack strategies depends on it. Therefore, the baseline information of the attacker is miscalculated and must be rectified. These statistical relationships are more easily understood using the concept of *functional-dependence graphs* (fd-graphs). We first review this concept briefly, in particular the implication of connectivity between nodes of a fd-graph on independence of corresponding random variables [41].

3.3.1 A Review of Functional-Dependence Graphs

A functional-dependence graph, or fd-graph, is a representation of a stochastic system in the form of a directed graph where random variables are represented as nodes of the graph and a directed edge between two nodes represents the existence of a direct causal statistical relationship between the corresponding random variables. If two random variables are independent then no directed edge connects them. Nodes with no incoming edges are known as source nodes. In a fd-graph $\mathcal{G}$ with disjoint subsets of nodes A, B, and C, the subset B is said to d-separate subsets A and C if no path exists from the nodes in A to nodes in C after the following manipulations have been performed

1: Create a sub-graph $\mathcal{G}'$ of $\mathcal{G}$ by considering only the links encountered while traveling backwards for any node in A, B, or C

2: Remove all the edges in $\mathcal{G}'$ outgoing from the nodes in B

3: Remove the directions from all remaining edges in $\mathcal{G}'$

fd-graphs simplify the analysis of statistical relationships between random variables, mainly due to the result that if B d-separates A and C, then A and C are independent conditioned on B; i.e. $I(A;C|B) = 0$ [41]. The fd-graphs for non-adaptive and adaptive attack strategies are presented in Figure 3.4a and 3.5a, respectively, which will be used to analyze the statistical relationship between different random variables in the leakage expression.

3.3.2 Statistical Relationships in Non-adaptive Strategies

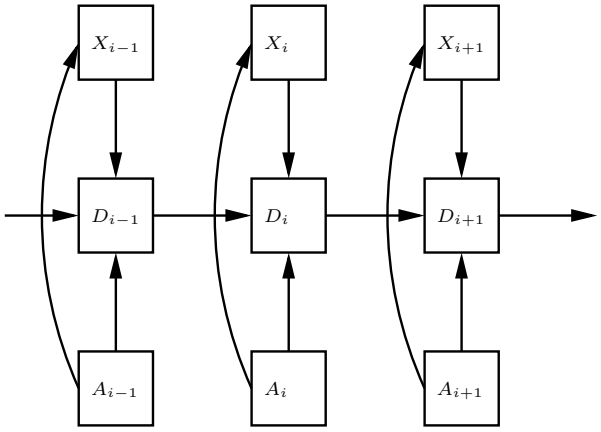
Figure 3.4b shows the fd-graph between X_i and A's conditioned upon A_i. It can be clearly seen that no edge exists between X_i and any A_j for $j \neq i$. This implies that conditioned on A_i, X_i is independent of all future and past A_j. Similar argument can be made for statistical relationship between X_i and X_j. Therefore,

$$H(X_i|X^{i-1}, A^n) = H(X_i|A_i)$$

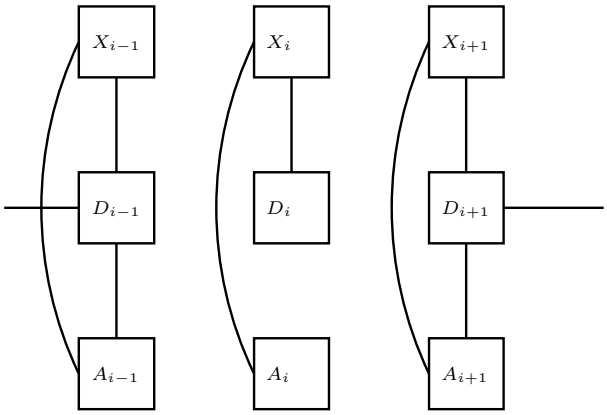
Similarly, from Figure 3.4c shows that X_i is independent of all other variables when conditioned on A_i, D_i, and D_{i-1}. This implies,

$$H(X_i|X^{i-1}, A^n, D^n) = H(X_i|A_i, D_i, D_{i-1})$$

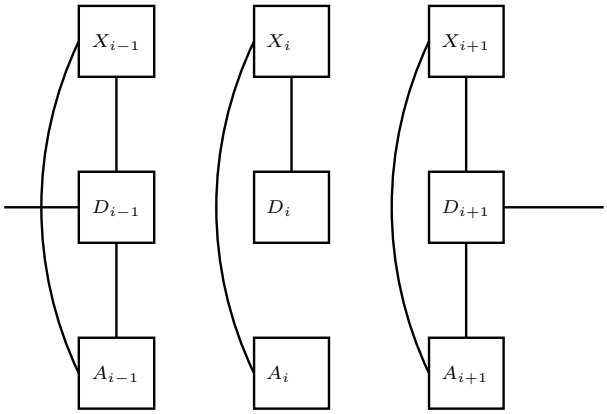
Next, we use the same tools to show that these relationships do not exist for adaptive attack strategies.



(a) Complete *functional-dependence* graph

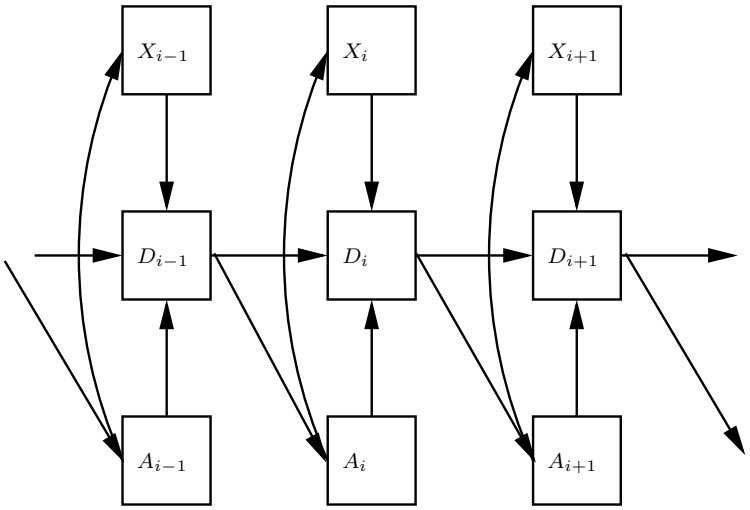


(b) Above fd-graph conditioned on A_i

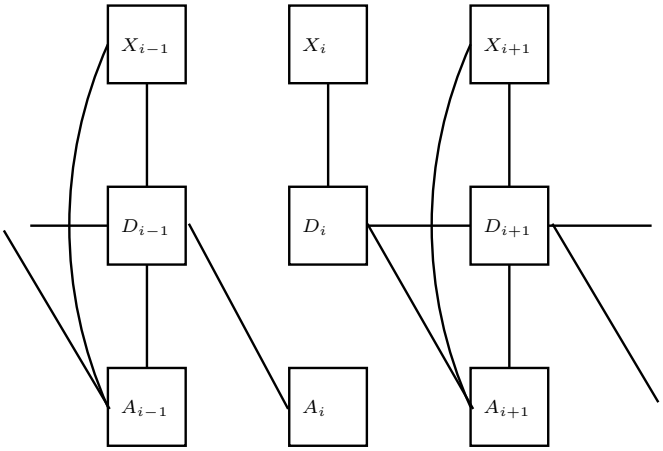


(c) Above fd-graph conditioned on A_i, D_i, D_{i-1}

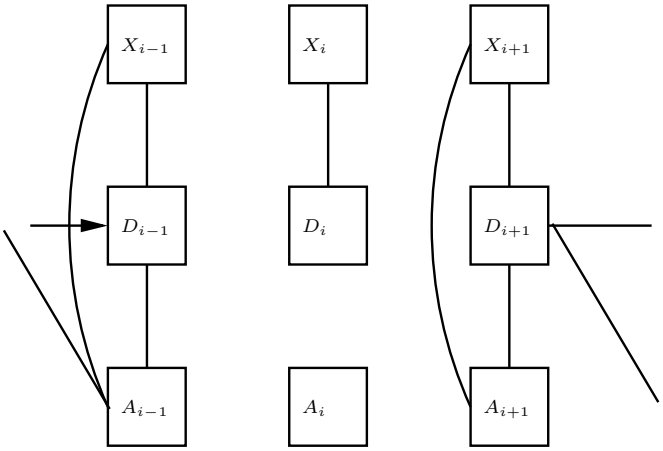
Figure 3.4: Strategies without feedback



(a) Complete *functional-dependence* graph



(b) Above *fd*-graph conditioned on A_i



(c) Above *fd*-graph conditioned on A_i, D_i, D_{i-1}

Figure 3.5: Strategies with first-order feedback

3.3.3 Statistical Relationships in Adaptive Attack Strategies

In the case of adaptive attack strategies, we first analyze the numerator; $H(X_i|X^{i-1}, A^n, D^n)$. As can been seen in Figure 3.5c, conditioned on A_i, D_i and D_{i-1}, X_i is independent of all other variables. This implies

$$H(X_i|X^{i-1}, A^n, D^n) = H(X_i|A_i, D_i, D_{i-1})$$

In contrast , Figure 3.5b that when conditioned on A_i, X_i and A_j for $j > i$ remains connected. This implies that these random variables are not independent and therefore,

$$H(X_i|X^{i-1}, A^n) \neq H(X_i|A_i)$$

This can be explained by the fact that in the case of adaptive strategies, knowledge of future inputs already includes knowledge of future outputs, and therefore entropy cannot be used to compute the baseline uncertainty of the attacker. This theme is recurrent in information-theoretic discussion on channels with feedback and specific concepts, such as *causally-conditioned entropy* and *directed information*, to deal with these scenarios. The definition we provided for the leakage of a general side channel in Chapter 1 incorporates these concepts already. Next, we use formalize the new definition of leakage that is appropriate for adaptive strategies and reduces to Gong *et al.*'s definition for non-adaptive strategies.

3.3.4 *Causal Leakage*

We introduce the notion of *causal leakage* for strategies with feedback that resolves this issue. To avoid an indirect use of side channel outputs in measuring the a-priori uncertainty of the attacker we employ the notions of *causally-conditioned* entropy [41]. We define the *causal leakage* $\mathcal{L}^c$ for feedback strategies:

$$\mathcal{L}^c = \lim_{n\to\infty} 1 - \frac{H(X^n||A^n, D^n)}{H(X^n||A^n)},$$

where $H(A^n||B^n) = \sum_i H(A_i|A^{i-1}, B^{i-1})$ is the entropy of the random sequence A^n *causally-conditioned* on the random sequence B^n [41]. Using this definition, we can easily find that

$$H(X^n||A^n) = \sum_i H(X_i|A_i)$$

and

$$H(X^n||A^n, D^n) = \sum_i H(X_i|A_i, D_i, D_{i-1})$$

The use of *causally-conditioned* entropy ensures the causal availability of information in the measurement of uncertainty and therefore, the metric is has better suitability. Now, we use this metric to find optimal adaptive attack strategies. [1]

[1]The use of the original definition $\mathcal{L}$ may overestimate the true information leakage of the system for adaptive strategies. This is because

$$H(X^n|T^n) \leq H(X^n||T^n)$$

3.4 Optimal Adaptive Strategies for Information Leakage

To identify optimal adaptive attack strategies, we first compute the *leakage* of a general adaptive attack strategy. The reason to analyze these strategies stems from the fact that in a side-channel attack, the attacker issues one of the inputs and observes the output. It is natural to consider the case when the attacker uses previous observations to decide future inputs; i.e. be adaptive. Not surprisingly, a number of real-world side-channel attacks are adaptive [6].

3.4.1 Leakage of a General Adaptive Strategy

A general adaptive attack strategy can be described a sequence of probability distributions, $\{p(a_i|a^{i-1}, d^{i-1})\}_{i=1}^{\infty}$, where the i^{th} distribution is used to choose the packet inter-arrival pattern of the i^{th} packet. This strategy is subject to an average probe rate constraint

$$\sum_i \sum_{a_i} a_i p(a_i|a^{i-1}, d^{i-1}) = \frac{1}{\lambda_2}$$

Strategies that use complete history in the determination of next input are impractical as their memory and computational requirements grow exponentially with the number of packets. Additionally, theoretical analysis of such strategies is not possible in a general case. We remove this hurdle first by showing that to find optimal adaptive strategies, the attacker does not need to use entire history. In fact, he only needs to store and use the latest side-channel output.

Theorem 8. *For any adaptive strategy that uses entire history, described as $p(a_{i+1}|a^i, d^i)$, there exists an adaptive strategy that only uses the queuing delay of the previous probe; i.e. $p(a_{i+1}|d_i)$ and achieves the same information leakage.*

Proof. The behavior of the queue can be modeled as a Markov Decision Process (MDP) where states are tuples (a_i, d_{i-1}) where the reward, measured in terms of mutual information between inputs and outputs, depends only on a_i and d_{i-1}. The transition between states depends only one the previous state and the action which the choice of the next input. For MDPs, a standard result known as the *dominance of Markov policies* [1] states that maximum reward is achieved by a Markovian strategy which chooses current action based solely on current state. □

Therefore, we will now restrict our analysis to strategies that can be specified as $p(a_i|d_{i-1})$. Furthermore, we limit the discussion to time-invariant strategies; i.e $p(a_i = a|d_{i-1} = d) = p(a|d)$. We can compute leakage by separately computing numerator and denominator. From

---

and therefore,

$$\mathcal{L} \geq \mathcal{L}^c$$

Intuitively, use of $H(X^n|T^n)$ inadvertently considers information imparted by future side channel inputs which within them contain information imparted by side channel outputs, thus understating the a-priori uncertainty of X^n.

the definition of causally-conditioned entropy

$$\begin{aligned} H(X^n||A^n) &= \sum_i H(X_i|X^{i-1}, A^{i-1}) \\ &= \sum_i H(X_i|A_i) \\ &= \sum_i \sum_{a_i} p(a_i) H(X_i|a_i) \\ &= \sum_i \sum_{a_i} p(a_i) H_B(\lambda_1, a_i) \\ &= \sum_i \sum_{a_i} p(d_{i-1}) p(a_i|d_{i-1}) H_B(\lambda_1, a_i) \end{aligned}$$

The limit $\lim_{n\to\infty} \frac{H(X^n||A^n)}{n}$ can be computed using Césaro's mean theorem if individual limiting probabilities exist. Since the choice of attack strategy is time-invariant, we only need to prove the existence of $\lim_{n\to\infty} p(d_{n-1})$. Similarly for the numerator,

$$\begin{aligned} H(X^n||A^n, D^n) &= \sum_i H(X_i|X^{i-1}, A^i, D^i) \\ &= \sum_i H(X_i|A_i, D_i D_{i-1}) \\ &= \sum_i \sum_{a_i, d_{i-1}} p(d_{i-1}) p(a_i|d_{i-1}) P_{EQ}(a_i, d_{i-1}) H(X_{a_i, d_{i-1}}) \end{aligned}$$

Again, to compute $\lim_{n\to\infty} \frac{H(X^n||A^n, D^n)}{n}$ using Césaro's mean theorem, we only need to show the existence of $\lim_{n\to\infty} p(d_{n-1})$. In essence, the above computations are similar to the analysis of non-adaptive strategies except that the probability distribution of probe inter-arrival times depends on the queuing delay of the probe. To show that $\lim_{n\to\infty} p(d_{n-1})$ exists, we only have to show that even in this case, queuing delay faced by a probe behaves as a first-order, irreducible Markov chain. If so, $\lim_{n\to\infty} p(d_{n-1})$ is simply the stationary distribution of the Markov chain.

Let D_n represent the state of system in the n^{th} time-slot. Then the transition probability, $p(D_{n+1} = g|D_n = h)$, denoted as p_h^g, can be derived as

$$p_h^g = \begin{cases} 0 & g > h+1 \\ \sum_{A=h-g+1}^{\infty} p(A|h) \binom{t}{h-g+1} (1-\lambda_1)^{h-g+1} \lambda_1^{A-(h-g+1)} & \text{for } g \in \{2, \cdots, h+1\} \\ \sum_{i=h+1}^{\infty} \sum_{A=i}^{\infty} p(A|h) \binom{t}{i} (1-\lambda_1)^i \lambda_1^{A-i} & \text{for } b = 1 \end{cases}$$

From the transition probabilities, it can be seen that the Markov chain is first-order, and irreducible because each state can be reached for every other state. Therefore, the stationary distribution of the Markov chain exists. Let $\pi(D)$ be the stationary probability of the queuing delay being D. Then, the leakage of a general adaptive strategy specified by the distribution $p(A_{i+1} = a|D_i = d)$ can be computed as

$$\mathcal{L}_c = 1 - \frac{\sum_{a,d} \pi(d)p(a|d)P_{EQ}(a,d)H(X_{a,d})}{\sum_{a,d} \pi(d)p(a|d)H_B(\lambda_1, a)}$$

3.4.2 Optimal Adaptive Strategies

The optimal strategy is found by solving the following linear program:

Maximize

$$\frac{\sum_{a,d} p(a,d)\left\{H_B(\lambda_1, a) - P_{EQ}(a,d)H(X_{a,d})\right\}}{\sum_{a,d} p(a,d)H_B(\lambda_1, a)}$$

Subject to

1: $0 \leq p(a,d) \leq 1$, for all a, d

2: $\sum_a p(a,d) = \pi(d)$ for all d

3: $\pi(d=h) = \sum_{g=h-1}^{\infty} \pi(d=g)p_g^h$, for all $h \in \{0, 1, \cdots\}$

4: $\sum_{a,d} ap(a,d) = 1/\lambda_2$

5: $\sum_d \pi(d) = 1$

6: $0 \leq \pi(d) \leq 1$, for all d

Due to the lack of a general form for the stationary distribution, the search for the optimal strategy that maximizes leakage using the above-mentioned linear program must treat the stationary distribution $\pi(d)$ and the joint distribution $p(a,d)$ as the control variables. The actual attack strategy can be determined as $p(a|d) = \frac{p(a,d)}{\pi(d)}$. Again, the objective function is a linear-fractional in terms of the control variables and the constraints are linear. Therefore, the linear program has a unique maximum value. Figure 3.6 illustrates the percentage enhancement achieved by optimal adaptive strategies over optimal non-adaptive strategies for the same bandwidth budget. Clearly, adaptive strategies achieve significant higher leakage and therefore, need to be part of a thorough side-channel analysis.

Similar to non-adaptive scenario, the stationary distribution that provides maximum leakage is the one that is biased towards a full queue. Intuitively, the adaptive attack strategy will ensure low probe inter-arrivals times when the queue is empty and large probe inter-arrival times when the queue is clogged. This ensures that queue remains full and therefore, the leakage is higher compared to non-adaptive strategies that have the same distribution on probe inter-arrival times irrespective of the queue lengths. For $\lambda_1 = 0.1$ and $\lambda_2 = 0.1$, the maximum leakage achieved

the optimal adaptive strategy is 0.43 as opposed to the maximum of 0.40 for non-adaptive strategies. The increment in leakage is smaller for lower λ_1 because the restriction on the average probe rate does not allow very short probe intervals even when the queue is empty. Figure 3.6 shows the performance enhancement in system leakage when full feedback is used by the attacker. It is clearly seen that a performance enhancement of nearly 28 % can be achieved for $\lambda_1 = 0.5$ and $\lambda_2 = 0.1$. Table 3.1 shows the optimal attack strategy for $\lambda_1 = 0.1$ and $\lambda_2 = 0.1$.

$D_{n-1}\|A_n$	1	2	3	4	5	$\cdots$
1	0.0281	0.0394	0.0360	0.0350	0.0342	$\cdots$
2	0.9982	0	0	0	0	$\cdots$
3	0.7639	0.2339	0	0	0	$\cdots$
4	0.8243	0.1716	0.0010	0	0	$\cdots$
5	0.8341	0.1586	0.0018	0	0	$\cdots$
6	0.8548	0.1131	0.0207	0.0038	0	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
45	0	0	0.9937	0.0024	0	$\cdots$
46	0	0	0	0.0141	0.9855	$\cdots$
47	0	0	0	0	0.0030	$\cdots$
48	0	0	0	0	0	$\cdots$
49	0	0	0	0	0	$\cdots$
50	0	0	0	0	0	$\cdots$

Table 3.1: Adaptive attack strategy for $\lambda_1 = 0.1$, $\lambda = 0.1$, $1 \leq a \leq 50$, and $1 \leq d \leq 50$

This example clearly validates the argument presented earlier. When the queue is empty, the inter-arrival time for the next packet is reduced to clog the queue. If the queue is clogged, the inter-arrival times may be increased without emptying the queue completely and ensuring low average probe rate.

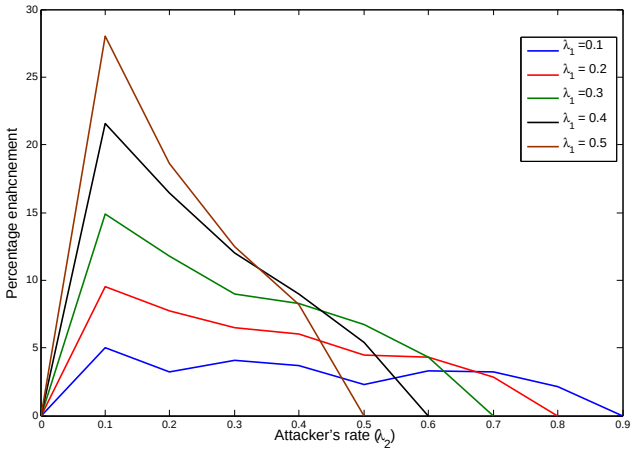


Figure 3.6: Percentage enhancement in system leakage due to feedback

3.5 Optimal Real-world Adaptive Attack Strategies

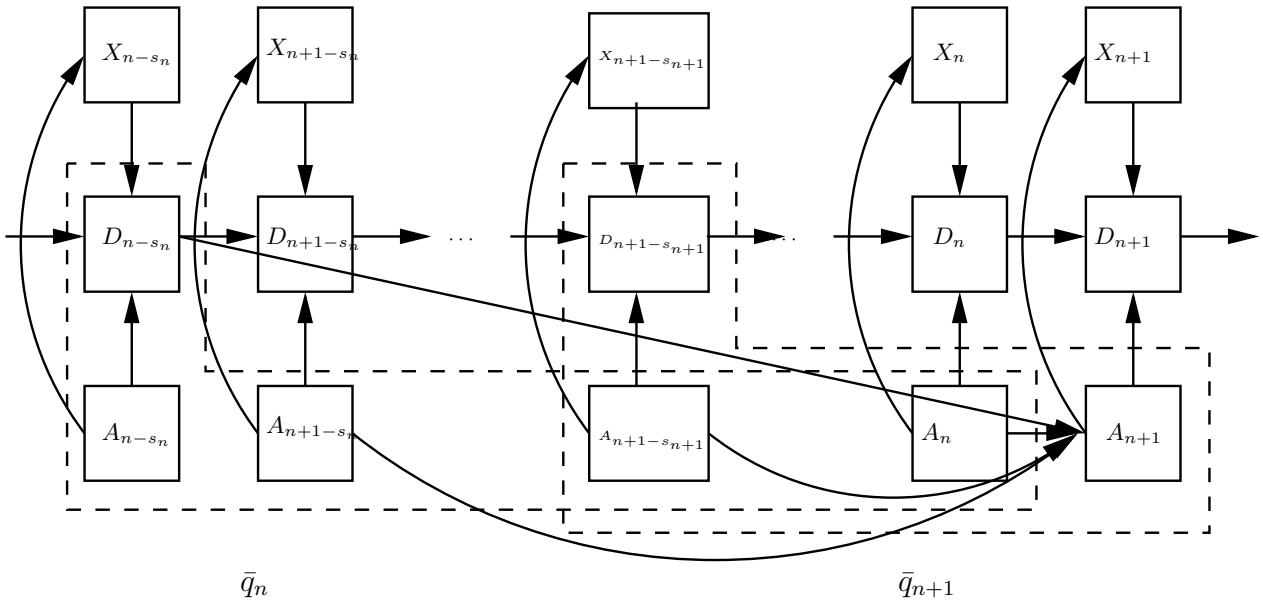
Adaptive strategies presented in the previous section allow the attacker to use all the previous information. That is, to decide the inter-arrival time for the n^{th}, the attacker can use inter-arrival times and queuing delays for all previous packets (from 1 to $n-1$). As the system is assumed to be causal, this is the maximum information that can be available to the attacker and therefore, optimal strategies that use this information remain globally optimal. Availability of this information, however, in not possible in practice. The time to decide the inter-arrival time of the n^{th} probe is when the $n-1^{th}$ probe is delivered to the scheduler. Since each probe requires at least one time-slot to be processed by the scheduler, the queuing delay of the $n-1^{th}$ probe cannot be known to the attacker eve if the queue is empty. Moreover, if the queue is clogged, a higher number of probes are stuck and their delays are unknown to the attacker. Realistically, the attacker can only use the queue delay of probes that have been served by the scheduler. However, he can use the inter-arrival times of all probes upto probe $n-1$.

Let s_{n-1} denote the *separation-of-index* between the $n-1^{th}$ probe and the latest probe which left the system when the $n-1^{th}$ probe entered the system; i.e. at time-slot t_{n-1}. That is, the index of the latest probe which left the system at time-slot t_{n-1} is $n-1-s_{n-1}$. As the inter-arrival time for the n^{th} probe, a_n, is decided at this moment, the attacker possesses the queuing delay information of all probes from 1 to $n-1-s_{n-1}$ but not for any later probe as they are still in the queue. Additionally, the attacker knows the inter-arrival time a_i for all $i \in \{1, \cdots, n-1\}$. Thus, the inter-arrival time of the current probe, a_n can be chosen as a function of the available information under the probability distribution $p\left(a_n | s_{n-1}, [a_j]_1^{n-1}, [D_j]_1^{n-1-s_{n-1}}\right)$ which specifies the attack strategy. Let $q_{n-1} \equiv s_{n-1}, [d_j]_1^{n-1}, [d_j]_1^{n-1-s_{n-1}}$ represent the collection of all parameters known to the attacker at time-slot t_{n-1}. We assume that the attacker maintains this as an internal state. We can analyze denominator and numerator of the leakage function similar to previous sections.

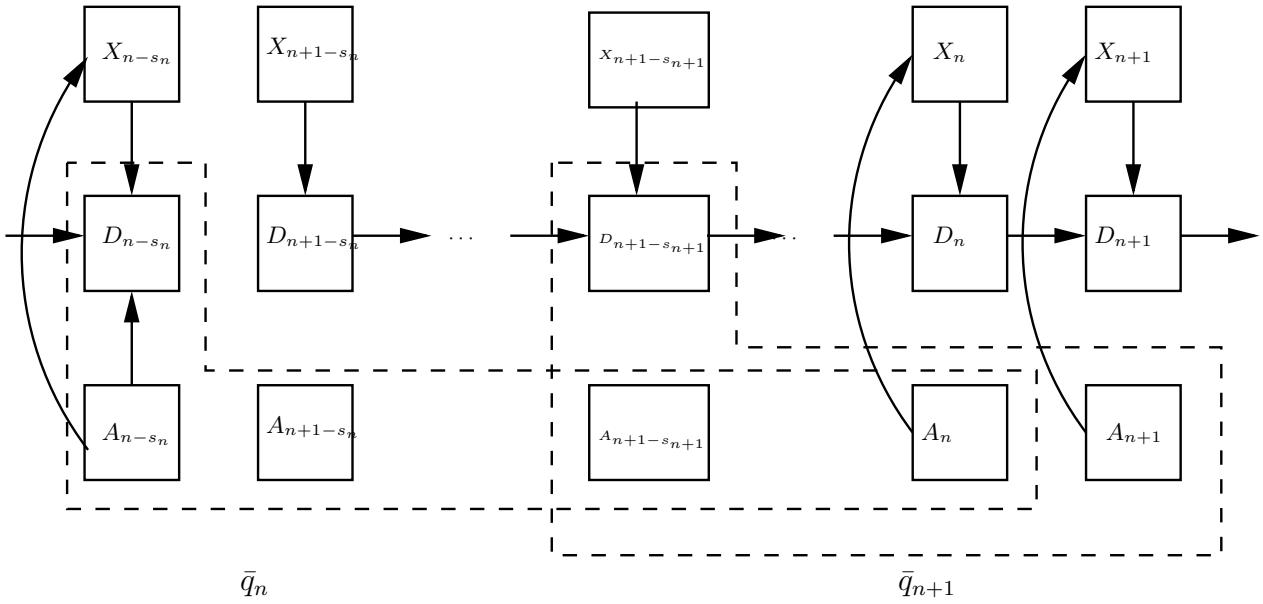
$$\begin{aligned}
H(X^n||A^n) &= \sum_n H(X_n|A_n) \\
H(X_n|A_n) &= \sum_{a_n} p(a_n) H(X_n|a_n) \\
&= \sum_{a_n, q_{n-1}} p(q_{n-1}) p(a_n|q_{n-1}) H(X_n|a_n) \\
&= \sum_{a_n, q_{n-1}} p(q_{n-1}) p(a_n|q_{n-1}) H_B(\lambda_1, a_n)
\end{aligned}$$

and

$$\begin{aligned}
H(X^n||A^n, D^n) &= \sum_n H(X_n|A_n, D_n, D_{n-1}) \\
&= \sum_{a_n, d_{n-1}} p(a_n, d_{n-1}) P_{EQ}(a_n, d_{n-1},) H(X_{a_n, d_{n-1}}) \\
&= \sum_{a_n, d_{n-1}, q_{n-1}} p(q_{n-1}) p(a_n, d_{n-1}|q_{n-1}) P_{EQ}(a_n, d_{n-1}) H(X_{a_n, d_{n-1}})
\end{aligned}$$



(a) Complete *functional-dependence* graph for practical adaptive strategies



(b) Above *fd*-graph conditioned on $\bar{q}_n$

Figure 3.7: Real-world adaptive strategies

A generic strategy that uses complete history has unrealistic memory requirement because with increasing n the state-space of q_n increases exponentially. We overcome this limitation by showing that to achieve maximal leakage, the attacker does not need to store entire past. Additionally, we prove a set of results that enable the identification of optimal real-world strategies while using partial feedback. Specifically,

Theorem 9. *Let* $\bar{q}_{n-1} \equiv s_{n-1}, [a_j]_{n-s_{n-1}}^{n-1}, d_{n-1-s_{n-1}}$. *Then,*
1) The optimal real-world adaptive strategy; one which is limited to use the delay of packets served by the scheduler, only requires to use the parameters $\bar{q}_{n-1}$ *to determine* a_n.
2) For the limit $\mathcal{L}^c = \lim_{n\to\infty} 1 - \frac{H(X^n||A^n,D^n)}{H(X^n||A^n)}$ *exists if limiting distributions* $\lim_{n\to\infty} p(\bar{q}_{n-1})$ *and*

$\lim_{n\to\infty} p(d_{n-1}|\bar{q}_{n-1})$ *must exist.*
3) For any time-invariant adaptive strategy; i.e. $p(a_n = a|\bar{q}_{n-1} = \bar{q}) = p(a|\bar{q})$, $\lim_{n\to\infty} p(\bar{q}_{n-1})$ *and* $\lim_{n\to\infty} p(d_{n-1}|\bar{q}_{n-1})$ *exist.*
4) $\bar{q}_n$ *for an irreducible and commuting Markov chain and therefore,* $\lim_{n\to\infty} p(\bar{q}_n)$ *can be computed as the stationary distribution of the Markov chain.*

Proof. The proof relies on the relationship between the queuing delay of two probes d_i and d_j, where $i < j$ without loss of generality. For these probes, we have

$$d_j = d_i + \sum_{k=i+1}^{j} (x_k + 1 - a_k)$$

Due to this, when analyzing the delay of a probe, the attacker only needs to store the delay of last-available probe and the inter-arrival times of all probes in-between. If the latest probe served by the system is $n - s_n$, then the delay of all probes $j > n - s_n$ is independent of past delay observations. This fact is illustrated in Figure 3.7a and 3.7b.

If the attacker stores the information $\bar{q}_n \equiv s_n, d_{n-s_n}, [a]_{n+1-s_n}^{n}$, then $\bar{q}_{n+1}$ depends only the choice of a_{n+1} and $\bar{q}_n$. The constituent terms of $H(X_i|A_i)$ and $H(X_i|A_i, D_i, D_{i-1})$ also depend only on these factors and the system can be modeled as a Markov Decision Process. The dominance of Markov policies immediately proves 1). 2), 3), and 4) can be proven using the relationship between delays. Detailed proofs are provided in the Appendix B.1. □

The optimal real-world strategy can be found by solving the linear program

Maximize

$$1 - \frac{\sum_{q_i, a_{i+1}, d_i} p(d_{i+1}|q_i) p(a_{i+1}, q_i) P_{EQ}(d_i, a_{i+1}) H(X_{d_i, a_{i+1}})}{\sum_{q_i, a_{i+1}} H_B(\lambda_1, a_{i+1})}$$

Subject to

1: $0 \le p(a_{i+1}, q_i) \le 1$, for all A_{i+1}, q_i

2: $\sum_{a_{i+1}} p(a_{i+1}, q_i) = \pi(q_i)$ for all q_i

3: $\pi(q_i = j) = \sum_{j=i-1}^{\infty} \pi(q_i = k) p_k^j$, for all $j \in \{0, 1, \cdots\}$

4: $\sum_{q_i, a} a_{i+1} p(A_{i+1}, q_i) = 1/\lambda_2$

5: $\sum_{q} \pi_q = 1$

6: $0 \le \pi_q \le 1$, for all q

Unlike previous scenarios, the number of variables in the linear program to compute optimal real-world adaptive strategies are prohibitively large. For $a \in \{1, 2, \cdots, |A|\}$, $s \in$

$\{1, 2, \cdots, |S|\}$, $d \in \{1, 2, \cdots, |D|\}$, the number of possible states $|Q_n| \approx |D| \times |A|^{|S|-1}$. The size of probability transition matrix, $p(A_{n+1}|Q_n)$, is $\mathcal{O}(|D| \times |A|^{|S|})$. For typical values of $|A| = |D| = 50$ and $|S| = 10$, the number of control variables in the linear program grow to $\approx 10^{18}$. This dimension is significantly high to be solved in reasonable time with any real-world attacker's computational power. Alternate methods and approximations may be required to solve this linear program realistically. Unfortunately, they are out of the scope of this thesis.

3.6 Conclusions

Results presented in this chapter demonstrate that quantitative modeling of side channels can even allow an attacker to develop optimal attack strategies. Optimal utilization of attack resources is important to launch real-world attacks against large systems such Tor. The enhancement in leakage due to such strategies can be very high to ignore. For the setup analyzed in this chapter, we were able to increase the leakage of the system upto 1000% over Geometric probing, despite not using feedback. With feedback, we were able to demonstrate a theoretical increase of upto 30% over non-adaptive strategies. While real-world optimal adaptive strategies require solving a large linear program, a determined attacker can either find the resources to it or find sub-optimal strategies that require less information.

At this point it is important to discuss the limitation of this analysis to a very specific packet arrival model from the user's side and the implication of a different model on leakage and optimal strategies. Optimal adaptive strategies identified in this chapter satisfy the intuition that adaptive strategies allow the attacker to probe at a faster rate when the queue is empty and probe slowly when queue is full. This intuition is likely to hold for alternative arrival process for user's packets. While more accurate models for real-world packet traffic exist, the leakage of the system under those models is likely to be higher due to the dependence between packet arrivals. Therefore, Bernoulli distribution for user's packet arrival forms the worst-case scenario from the attacker's point-of-view.

Chapter 4

Side Channels in Cryptographic Algorithms

Asymmetric cryptosystems, such as RSA [54] and Diffie-Hellman Key-Exchange [15], require computationally-intensive modular exponentiation/multiplication operations. This limits their applicability on devices with low processing capabilities or in services which require significant data processing rates. To address this issue, several algorithms have been developed to perform modular multiplications efficiently in hardware. One of the most efficient and widely-used algorithms, named Montgomery Multiplication (MM) (Algorithm 4), was devised by Peter Montgomery [46]. Montgomery Multiplication replaces computationally-expensive divisions with the modulus M to multiplications/divisions with the Montgomery reduction parameter R. R is chosen to be a power of two; i.e. $R = 2^x$ for some integer x, and therefore, multiplications and divisions with R are computationally-inexpensive bit-shifts. However, Montgomery Multiplication occasionally requires an extra reduction step depending on the relative values of the multiplicands and modulus, which causes a discrepancy in the amount of time required for the multiplication. This leads to the creation of a side channel which has been exploited to break several cryptosystems [37],[6],[13].

Goal of timing attacks against modular exponentiation-based cryptosystems is to either learn the secret key/exponent in the case when modulus is known or to learn the modulus in case it is unknown. The second scenario arises under RSA implementation with Chinese Remainder Theorem which uses RSA prime factors to perform exponentiation. In this analysis, we compute the leakage of secret key/exponent-bits through this side channel and the accuracy of the attacker in learning the secret modulus for the latter case. Our quantitative analysis is based on the theoretical model developed by Schindler and others [58],[57],[56]. This model relies on the computation of probability of an extra reduction in individual Montgomery Multiplications. Using this modeling, we make the following contributions in this analysis:

- ***Reliability rate for estimation of unknown prime modulus:*** We compute the reliability rate of an attacker that aims to learn the modulus used in a modular exponentiation. This scenario arises in RSA implementation that use CRT. In such cases, the modulus is one of the prime factors of the RSA modulus. We compute the reliability rate for this scenario and discover the relationship between reliability rate, Montgomery reduction parameter, and the size of the RSA prime.

- ***Key/exponent leakage in the Montgomery Multiplication routine:*** We develop a new

model for the timing side channel in the Montgomery Multiplication routine and provide lower and upper bounds on the leakage of the routine for non-adaptive strategies. We show that the leakage of the routine decreases with increasing Montgomery parameter.

- ***Analysis of countermeasures and their performance trade-offs:*** Lastly, we employ our leakage model to quantify the efficacy of two popular countermeasures against timing attacks, namely, exponent blinding and caching. We compute the reduction in leakage of the Montgomery Multiplication routine in the presence of each countermeasure and their performance trade-offs with resource budgets. Importantly, we identify the conditions under which one countermeasure outperforms the other.

First, we briefly review the specifics of exponentiation algorithms and the Montgomery Multiplication routine.

4.1 Preliminaries

One of the key operations in modular exponentiation based cryptographic algorithms, such as RSA and DH, is to compute $y^d(mod\ M)$, where y is the ciphertext, d is the exponent, and M is the modulus. This operation is performed by a series of multiplications, where the multiplicands depend on the exponent bit. A typical modular exponentiation is performed using the *square-and-multiply* algorithm:

Data: input: y, exponent: d, modulus: M
Result: $c = y^d(mod\ M)$
$temp := y$;
for i=2:$|d|$ **do**
  $temp := temp^2(mod\ M)$;
  if $b_i == 1$ **then**
    $temp = temp * y(mod\ M)$;
  end
end

Algorithm 4: *Square-and-multiply* algorithm for exponentiation

If the exponent bit is 0, the only operation performed is squaring of the $temp$ value. If the exponent bit is 1, an additional multiplication with the ciphertext y is performed. Each of these multiplications is performed using an optimized modular multiplication algorithm: Montgomery Multiplication (Algorithm 4). This algorithm succeeds in performing modular multiplication efficiently because it transforms multiplication/division operations under an odd-modulus M with similar operations under another base R which is chosen to be a power of two; i.e. $R = 2^x$, for some x. Such operations are simple bit-shifts and therefore, computationally fast. Let R^{-1} represents the multiplicative inverse of R modulo M and M^* is an integer such that $RR^{-1} - MM^* = 1$. Montgomery Multiplication invokes two transforms: $\Psi(a) = aR(mod\ M)$ and $\Psi^{-1}(a) = aR^{-1}(mod\ M)$.

To multiply two numbers a and b, the Montgomery Multiplication routine takes $\Psi(a)$, $\Psi(b)$ as inputs and outputs $\Psi(c)$, where $c = a \times b(mod\ M)$.

An important advantage of using the Montgomery Multiplication routine for modular multiplication is that the output is already in the form suitable for the next multiplication. Therefore,

Data: $\Psi(a)$, $\Psi(b)$, M
Result: $\Psi(c)$, where $c = a \times b(mod\ M)$
Step 1: $z := \Psi(a)\Psi(b)$;
Step 2: $z' := (z(mod\ R)M^*)(mod\ R)$;
Step 3:$\Psi(c) := \frac{(z+z'M)}{R}$;
if $\Psi(c) \geq M$ **then**
  Step 4: $\Psi(c) := \Psi(c) - M$;
end

Algorithm 5: The Montgomery Multiplication routine

the transforms Ψ and Ψ^{-1} are invoked only once during a modular exponentiation. Step 4 of the Montgomery Multiplication routine is known as the extra-reduction step and is the cause of timing variations in the Montgomery Multiplication routine. Since a modular exponentiation is simply a series of Montgomery Multiplications, the timing discrepancy of a modular exponentiation depends on the number of extra reductions in these Montgomery Multiplications. Next, we review the probability of an extra reduction in each individual Montgomery Multiplication and stochastic behavior of total timing for a modular exponentiation.

4.2 Stochastic Modeling for Timing Side Channel

The building block of the stochastic modeling of modular exponentiation is the probability of observing an extra reduction in each individual (or constituent) Montgomery Multiplication operation. This probability depends on the value of the $temp$ variable, the ciphertext y, and the current bit of the exponent b as it decides whether is multiplication is $temp^2$ or $temp \times y$. We first start by computing the probability of an extra reduction in each case. This computation is based on results developed by Schindler [57, 58], Sato *et al.* [56], and Walter [64].

4.2.1 Probability of an Extra Reduction in a Single Modular Multiplication

The following lemma describes the conditions under which an extra reduction in Montgomery Multiplication is required.

Lemma 1. *[56, 58, 63, 64]*
i. a) Montgomery Multiplication of ciphertext y and $temp$ modulo M requires an extra reduction step iff

$$\frac{y \times temp}{RM} + \frac{(y \times temp \times M^*)(mod\ R)}{R} \geq 1$$

i. b) From square-and-multiply *and the Montgomery Multiplication routine, we have*

$$\frac{temp_i}{M} = \left(\frac{y\ temp_{i-1}}{M^2} \frac{M}{R} + \frac{y\ temp_{i-1} M^*(mod\ R)}{R} \right) (mod 1)$$

An extra reduction is carried out iff

$$\frac{temp_i}{M} < \frac{y}{M}\frac{temp_{i-1}}{M}\frac{M}{R}$$

Similarly,
ii.a) Montgomery Multiplication of $temp$ *and* $temp$ *modulo* M*; i.e. squaring, requires an extra reduction step iff*

$$\frac{temp^2}{RM} + \frac{(temp^2 \times M^*)(mod\ R)}{R} \geq 1$$

ii.b) From square-and-multiply *and the Montgomery Multiplication routine, we have*

$$\frac{temp_i}{M} = \left(\frac{temp_{i-1}^2}{M^2}\frac{M}{R} + \frac{temp_{i-1}^2 M^*(mod\ R)}{R}\right)(mod 1)$$

An extra reduction is carried out iff

$$\frac{temp_i}{M} < \frac{temp_{i-1}^2}{M^2}\frac{M}{R}$$

Proof. The proof is a direct implication of Step 4 in Algorithm 2 and properties of modulo 1. □

As repeated multiplications are performed for an exponentiation, the behavior of $temp_i$ can be modeled as a random-variable which equi-distributed on Z_m. This implies that the occurrence of an extra reduction in a Montgomery Multiplication is also random. We have,
Lemma 2. *[58] a) Let* $temp$ *be random variable equi-distributed on* Z_M *and* y *be a fixed ciphertext. Then,*

$$\textit{Prob(extra reduction in } y \times temp(mod\ M)) = \frac{y(mod\ M)}{2R}$$

b) Let $temp$ *be random variable, equi-distributed on* Z_M*. Then,*

$$\textit{Prob(extra reduction in } temp^2(mod\ M)) = \frac{M}{3R}$$

Proof. The proofs rely on the fact that the terms $\frac{temp}{M}$, $\frac{(y \times temp \times M^*)(mod\ R)}{R}$, and $\frac{(temp^2 \times M^*)(mod\ R)}{R}$ behave like i.i.d. random variables, uniformly-distributed over $(0, 1)$. Detailed steps can be found in Appendix C.1. □

Let $S_i \equiv \frac{temp_i}{M}$ and $W_i \in \{0, 1\}$ be a random variable that represents the occurrence of an extra reduction ($w_i = 1$) or not ($w_i = 0$). $temp_i$ in a modular exponentiation behaves like an i.i.d. random variable equi-distributed over Z_M [56, 57]. Therefore, S_i behaves like an i.i.d. random variable uniformly-distributed over the set $(0, 1)$. Using the conditions presented in Lemma 1, we compute the conditional probability distribution $p(w|b, y)$ as:

The stochastic model presented in Table 4.1 forms the basis of stochastic model for total decryption timing and leakage analysis in Section VI. Next, we present the stochastic modeling of total timing behavior of a modular exponentiation.

$P(W\|B,Y)$	$B_i = 0$	$B_i = 1$
$W_i = 0$	$P\left(S_i \geq \frac{S_{i-1}^2 M}{R}\right)$	$P\left(S_i \geq \frac{S_{i-1} y_i M}{R}\right)$
$W_i = 1$	$P\left(S_i < \frac{S_{i-1}^2 M}{R}\right)$	$P\left(S_i < \frac{S_{i-1} y_i M}{R}\right)$

Table 4.1: Conditional probability distribution, $P(W|B,Y)$

4.2.2 Timing Behavior of Modular Exponentiation

For the same exponent d, the number of squarings and multiplications performed in an exponentiation is the same. However, the probability of observing an extra reduction in each Montgomery Multiplication depends on both y and M. This probability also depends on whether the Montgomery Multiplication is a squaring or a multiplication. Let $|d|$ and d_1 denote the total number of bits and the total number of 1's in the in the binary representation of the exponent d, respectively. Then, in the modular exponentiation $y^d (mod\ M)$, a total of $|d|$ squarings and d_1 multiplications are performed. We assume that each of these operations requires c units of times. An additional c_{ER} units of time are required if an extra reduction is performed. The conditions and probabilities of observing an extra reduction in these operations are described in Lemma 1 and Table 4.1.

Let, $W_i \in \{0, 1\}$ denote the requirement of an extra reduction for the i^{th} Montgomery Multiplication, then the total time required to compute $y^d (mod\ M)$, $T(y)$ is given by

$$T(y) = \sum_{i=1}^{|d|} (c + c_{ER} w_i) + \sum_{i=1}^{d_1} (c + c_{ER} w_i).$$

From Table 1, it can be seems that the probability of $w_i = 0/1$ depends on b_i, y, s_i and s_{i-1}. Therefore, random variables W_i's are neither independent nor identically distributed as their distribution depends on the operation being a squaring or a multiplication. They are also dependent on the value of the previous state of the algorithm; i.e. S_{i-1}. Still, the total timing of a decryption, $T(y)$, is the sum of a large number of dependent random variables. Its p.d.f. can be computed using the central limit theorem [32].

Theorem 10. *[57] The total time $T(y)$ to compute $y^d (mod\, p)$ can be represented by a normally-distributed random variables $\mathcal{N}(\mu, \sigma^2)$, where*

$$\mu = c(|d| + d_1) + c_{ER}|d|\frac{M}{3R} + d_1\frac{y}{2R}$$

and

$$\begin{aligned} \sigma^2 \;=\; & c_{ER}^2 \left\{ |d| \left(\frac{M}{3R} - \left(\frac{M}{3R}\right)^2 \right) + d_1 \left(\frac{y}{2R} - \left(\frac{y}{2R}\right)^2 \right) \right. \\ & \left. +2(d_1 - 1)Cov_{SM} + 2d_1 Cov_{MS} + 2(|d| - d_1)Cov_{SS} \right\} \end{aligned}$$

where,

$$\begin{aligned} Cov_{SM} &= \frac{1}{10}\frac{M^2}{R^2}\frac{y}{R} - \frac{M}{3R}\frac{M}{2R} \\ Cov_{MS} &= \frac{1}{12}\frac{y^3}{R^3}\frac{M}{R} - \frac{M}{3R}\frac{M}{2R} \\ Cov_{SS} &= \frac{1}{21}\frac{M^4}{R^4} - \frac{M}{3R}\frac{M}{2R} \end{aligned}$$

Proof. A consequence of central limit theorem for weakly-dependent variables. Detailed steps are presented in Appendix C.2. □

4.3 Reliability Rate for Timing Attacks on Modular Exponentiation with Unknown Exponent

In certain cases, the goal of the attacker is to learn the modulus being used in a modular exponentiation. For example, this scenario occurs in RSA implementations that use the Chinese Remainder Theorem (CRT) (Algorithm 3). CRT is used because it reduces the computation of $y^d(mod\ M)$, with two exponentiations albeit with smaller exponents and modulus, which are unknown. As the modulus is unknown, timing attacks that reveal the secret key are not possible. However, the secret modulus itself can be learned as computation times depend on it. Knowledge of exponentiation modulus in this scenario allows the attacker to break RSA as the modulus is one of the prime factors of the RSA modulus. We compute the optimal reliability rate of an attacker in learning this information.

4.3.1 Problem Formulation

We formulate the problem of estimating the unknown modulus as a multi-hypothesis testing problem. Let the modulus, M, be chosen from an ordered set of possible modulus $\mathcal{M}$. The attacker sequentially sends ciphertexts y_i to the oracle which then decrypts the ciphertext with a secret exponent, d, and modulus, M. The side-channel outputs the time taken for this operation, $T(y_i)$, to the attacker. The attacker uses his previously issued inputs and observed outputs to decide the next ciphertext through a function $y_i = f(y^{i-1}, T^{i-1}(y))$; i.e. is adaptive. After sending n inputs and observing the corresponding outputs, the attacker produces an estimate of the underlying modulus, $\hat{M}$, using an estimator $g(y^n, T^n(y))$ The attacker makes an error if $\hat{M} \neq M$. The optimal reliability rate of the attacker, R^*_{opt} is measured as:

$$R^*_{opt} = \max_{f,g} \lim_{n\to\infty} \frac{-\log_2 P[\hat{M} \neq M]}{n}$$

For given y_i and M, $T(y_i)$ behaves like a normally-distributed random variable, where the mean, $\mu_{y_i,M}$, and variance, $\sigma^2_{y_i,M}$, are described in Theorem 10. Since the difference between variance of $T(y)$ for different parameter values is not significant, we assume it to be a constant value σ^2. With the model of this hypothesis test available, we use Naghshvar and Javidi's work on the optimal error-exponent for a general multi-hypothesis testing problem [49]. This computation consists of the following steps.

i) For each modulus $M_i \in \mathcal{M}$ and a given probability distribution on the input ciphertexts, $Q(y)$, minimum expected KL divergence is computed with respect to all $M_j \neq M_i$. That is,

$$R(i,Q) = \min_{j \neq i} \sum_y q(y) D\left[\mathcal{N}(\mu_{y,M_i}, \sigma^2) || \mathcal{N}(\mu_{y,M_j}, \sigma^2)\right]$$

ii) Next, a probability distribution $Q^*(i)$ is computed that minimizes $R(i,Q)$. The corresponding expected KL-divergence, $\bar{R}(i)$ is computed as,

$$\bar{R}(i) = \max_Q R(i,Q)$$

iii) Finally, the harmonic mean of all $\bar{R}(i)$ is computed over all $M_i \in \mathcal{M}$. This value is the optimal error exponent R^*_{opt}

$$R^*_{opt} = \frac{1}{\sum_i \frac{1}{\bar{R}(i)}}$$

R^*_{opt} is the maximum error-exponent that can be achieved by any sequential and adaptive adversary and therefore, is the optimal reliability rate.

We start by computing $R(i,Q)$ for the $M_i \in \mathcal{M}$. The KL divergence between two normally-distributed random variables with different means and same variance, $\mathcal{N}(\mu_a, \sigma^2)$ and $\mathcal{N}(\mu_b, \sigma^2)$ can be computed as:

$$D\left[\mathcal{N}(\mu_a, \sigma^2) || \mathcal{N}(\mu_b, \sigma^2)\right] = \frac{(\mu_a - \mu_b)^2}{2\sigma^2}$$

Lemma 3. *For $M_i \in \mathcal{M}$ and given probability distribution, Q, on y, we have*

$$R(i,Q) = \frac{|d|^2}{2\sigma^2 R^2} \min\{R_l(i,Q), R_u(i,Q)\}$$

where,

$$\begin{aligned}
\Delta_{i-1} &= M_i - M_{i-1} \\
\Delta_{i+1} &= M_i - M_{i+1} \\
R_l(i,Q) &= \Delta_{i-1}^2 \left[\frac{1}{9} - \frac{5}{48} Q(y > M_i))\right] + \left[\frac{M_{i-1}\Delta_{i-1}}{6} + \frac{M_{i-1}^2}{16}\right] Q(M_{i-1} < y < M_i) \\
R_u(i,Q) &= \Delta_i^2 \left[\frac{1}{9} - \frac{5}{48} Q(y > M_{i+1}))\right] + \left[\frac{M_i\Delta_i}{6} + \frac{M_i^2}{16}\right] Q(M_i < y < M_{i+1})
\end{aligned}$$

Detailed proof of Lemma 3 is provided in Appendix C.3. The importance of Lemma 3 is that it reduces the search of minimum KL divergence over the entire set $\mathcal{M}$ to two elements of the set, namely the lower and higher elements with respect to M_i. To compute the probability distribution Q that maximizes $R(i,Q)$, we need an ordered list of elements in $\mathcal{M}$ that can allows to compute the preceding and succeeding elements of M_i. This implies that the analysis has be to be performed separately for different $\mathcal{M}$. We compute the reliability rate when the modulus is prime number of a certain size, as this scenarios arises in RSA with CRT.

Analysis for prime modulus: Although there is no maintained (or maintainable) list of primes of a given size: $|d|$-bits, we still proceed with this analysis by approximating the prime gaps between two consecutive $|d|$-bit primes with the average prime gap for such primes.

Lemma 4. *The number of primes of length $|d|$-bits is $\frac{(2^{|d|}-2^{|d|-1})}{|d|\log_2 e}$, and their average prime gap is $|d|\log_2 e$.*

Proof. A corollary of the prime counting theorem [31]. □

Assuming, Δ_{i-1}, $\Delta_i \approx |d|\log_2 e$, we get

Lemma 5.

$$\bar{R}(i) \equiv \max_{Q} R(i,Q) \approx \frac{|d|^2 p_i^2}{32\sigma^2 R^2}.$$

where p_i is the i^{th} prime of length $|d|$ bits.

Detailed proof of Lemma 4 can be found in Appendix C.4. Finally, substituting $\bar{R}(i)$, we have

Theorem 11. *The optimal error-exponent, R^*_{opt} for detecting underlying RSA primes of $|d|$ bits while using Montgomery reduction parameter, R, can be computed as*

$$R^*_{opt} = \frac{|d|^2}{32\sigma^2 R^2}\left\lfloor \frac{|\mathcal{P}_{|d|}|}{\sum\limits_{p\in\mathcal{P}_{|d|}} \frac{1}{p^2}} \right\rfloor$$

where $\mathcal{P}_{|d|}$ is the set of all primes of length $|d|$-bits.

Theorem 11 computes the optimal reliability rate that can be achieved by an attacker who estimate the underlying prime modulus. We can draw two conclusions from this analysis. One, that the optimal reliability rate is positive and therefore, the attacker can learn the secret prime. Second, that the optimal reliability rate is inversely proportional to the Montgomery reduction parameter, R. Since R has to be larger than the modulus, system designers should select the value of the R as the largest power of 2 that is permissible by the underlying computing architecture. A limitation of this analysis is that it needs to be performed separately for different $\mathcal{M}$. To perform a general analysis, we compute the leakage of the Montgomery Multiplication routine itself in the net section.

4.4 Leakage of the Montgomery Multiplication Routine

The root cause of the timing side channel in modular exponentiation-based cryptosystems is the inconsistency of extra reductions in the underlying multiplication routine; i.e. the Montgomery Multiplication routine. Most existing analyses, including that performed in the previous section, focus on the analysis of specific attacks. However, the analysis of information leakage in the Montgomery Multiplication routine itself can be of great importance as it can allow generalization of analysis for all cryptographic algorithms that employ the Montgomery Multiplication routine. Simultaneously, this analysis can allow system designers to implement countermeasures directly in the implementation of the Montgomery Multiplication routine and also analyze their performance. In this section, we develop a model for the timing channel present in the Montgomery Multiplication routine and compute its leakage.

4.4.1 A Side-channel Model for the Montgomery Multiplication Routine

The side-channel model for the Montgomery Multiplication routine is developed in line with the general side-channel model presented in Chapter 1. The algorithm is abstracted as a discrete-time, two-input-single-output system. In every time-slot, the user issues a binary input, $b_i \in \{0, 1\}$, which is the current exponent bit. The attacker issues a corresponding input ciphertext, $y_i \in Z_M$, to the side channel. The Montgomery Multiplication routine maintain a state $S_i \equiv \frac{temp_i}{M}$ in each time-slot. The variable $temp_i$ is the temporary variable maintained by a *square-and-multiply* algorithm. Depending on b_i, the Montgomery Multiplication routine is used to either perform $temp_i^2(mod\ M)$ (for $b_i = 0$) or $y_i \times temp_i(mod\ M)$ (for $b_i = 1$). The side channel produces a corresponding output bit $w_i \in \{0, 1\}$ with the attacker observes. $w_i = 0$ implies that the i^{th} Montgomery Multiplication operation did not require an extra reduction, whereas $w_i = 1$ implies that an extra reduction was required. Figure 4.1 illustrates the side-channel model.

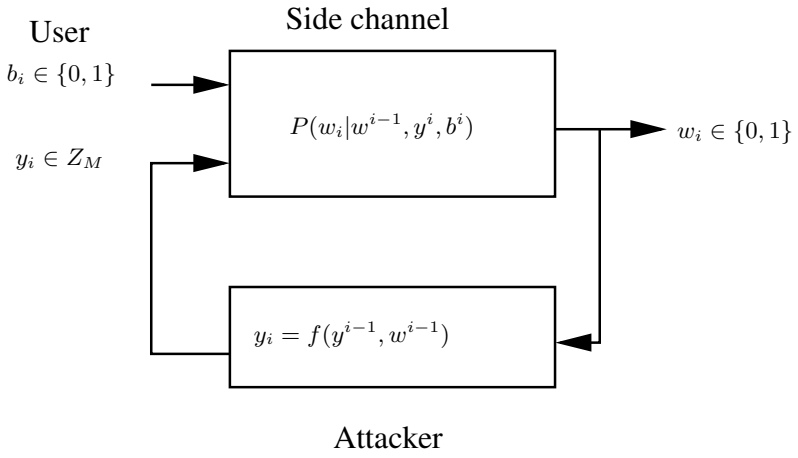


Figure 4.1: Timing side channel in the Montgomery Multiplication routine

The stochastic relationship between the side-channel inputs, output, and internal state variables is as specified in Table 4.1, where S_i's behave as i.i.d. random variables, uniformly-distributed over $(0, 1)$. The goal of the attacker is to estimate the user's input bit-sequence B^n, given the knowledge of his inputs, Y^n and the side-channel outputs, W^n. The attacker may be adaptive; i.e. choose his next input, y_i, based on previously issued inputs, y^{i-1}, and observed outputs, w^{i-1}, using a stochastic function $p(y_i|y^{i-1}, w^{i-1})$. If the probability distribution on the attacker's inputs $p(y_i)$ is independent of the past, then the attacker is said to be non-adaptive. The leakage of the Montgomery Multiplication routine, $\mathcal{L}_{MM}$, for an attack strategy, $p(y_n|y^n, w^n)$ is defined as

$$\mathcal{L}_{MM} = \lim_{n\to\infty} 1 - \frac{H(B^n||Y^n, W^n)}{H(B^n)}$$

Next, we compute the side-channel leakage of the Montgomery Multiplication routine for non-adaptive strategies under this model.

4.4.2 Bounds on the Leakage of the Montgomery Multiplication Routine for Non-adaptive Strategies

First, we compute a lower-bound on the information leakage of the Montgomery Multiplication routine to show that leakage is non-trivial; i.e. $\mathcal{L}_{MM} \geq 0$. This suffices to demonstrate that information is leaked through this side channel at a positive rate.

Theorem 12. *For a non-adaptive strategy, specified by the probability distribution $p(y)$*

$$\mathcal{L}_{MM} \geq 1 - \mathbb{E}_y \left[\left(\frac{y}{4R} + \frac{M}{6R} \right) H \left(\frac{\frac{y}{4R}}{\frac{y}{4R} + \frac{M}{6R}} \right) + \left(1 - \frac{y}{4R} - \frac{M}{6R} \right) H \left(\frac{1 - \frac{y}{4R}}{1 - \frac{y}{4R} + \frac{M}{6R}} \right) \right]$$

Proof. To compute a lower bound on $\mathcal{L}_{MM}$, we compute an upper-bound on $H(B_n|B^{n-1}, Y^n, W^n)$. We have $H(B^n||Y^n, W^n) = \sum_n H(B_n|B^{n-1}, Y^n, W^n)$. Using Césaros's mean theorem [48],

$$\lim_{n\to\infty} \frac{H(B^N||W^n, Y^n)}{n} = \lim_{n\to\infty} H(B_n|B^{n-1}, Y^n, W^n)$$

Further, we have

$$\begin{aligned} H_{con}(n) &\leq H(B_n|W_n, Y_n) \\ &= \sum_{y_n, w_n} p(y_n)p(w_n|y_n)H(B_n|w_n, y_n) \\ &= \sum_{y_n} p(y_n) \left[\left(\frac{y_n}{4R} + \frac{M}{6R} \right) H \left(\frac{\frac{y_n}{4R}}{\frac{y_n}{4R} + \frac{M}{6R}} \right) + \left(1 - \frac{y_n}{4R} - \frac{M}{6R} \right) H \left(\frac{1 - \frac{y_n}{4R}}{1 - \frac{y_n}{4R} - \frac{M}{6R}} \right) \right] \end{aligned}$$

□

Here, we have assumed that the attacker's input strategy is non-adaptive. However, the maximum leakage achieved by adaptive strategies can only be higher than that achieved by non-adaptive strategies and therefore, the lower-bound on leakage still holds.

An upper-bound on the leakage of a non-adaptive strategy is computed next. Upper-bounds can be used to evaluate performance of countermeasures as will be shown in the next section.

Theorem 13. *The leakage of the Montgomery Multiplication routine, $\mathcal{L}_{MM}$, for a given probability distribution $P(y)$ can be computed as*

$$\mathcal{L}_{MM} \leq \sum_y p(y) \left[\frac{y}{2R} - \frac{M}{3R} - \frac{y^3}{3M^2R} \right]$$

Proof. Again, let $H_{con}(B_n) = H(B_n|B^{n-1}, W^n, Y^n)$. Then,

$$\begin{aligned} H_{con}(B_n) &\overset{(a)}{\geq} H(B_n|B^{n-1}, W^n, Y^n, S^n) \\ &\overset{(b)}{=} H(B_n|W_n, Y_n, S_{n-1}, S_n) \end{aligned}$$

(a): Conditioning can only reduce entropy.
(b): Given w_n, s_n, s_{n-1}, and y_n, b_n is independent of the past (Table 4.1).

From the stochastic relationship between these variables, presented in Table 4.1, one can see that $w_n = 0$, irrespective of the value of b_n, iff $s_n > \max\{\frac{s_{n-1}^2 M}{R}, \frac{s_{n-1} y_n M}{R}\}$. Hence, if this relation is satisfied, the attacker cannot guess the user's bit and $H_{con}(B_n) = 1$. Similarly, if $s_n \leq \min\{\frac{s_{n-1}^2 M}{R}, \frac{s_{n-1} y_n M}{R}\}$, $w_n = 1$ irrespective of the value of b_n.

Therefore, the attacker can only learn the user's bit with certainty is $\min\left\{\frac{s_{n-1}^2 M}{R}, \frac{s_{n-1} y_n M}{R}\right\} < s_n \leq \max\left\{\frac{s_{n-1}^2 M}{R}, \frac{s_{n-1} y_n M}{R}\right\}$. For every other case, $H_{con}(B_n) = 1$. This relationship can be simplified by considering two different ranges of s_{n-1}: a) $s_{n-1} \in \left(0, \frac{y_n}{M}\right)$ and b) $s_{n-1} \in \left(\frac{y_n}{M}, 1\right)$.

$$
\begin{aligned}
H_{con}(B_n) &\geq 1 - \sum_y p(y) \left[\int_0^{\frac{y}{M}} \int_{\frac{s_{n-1}^2 M}{R}}^{\frac{s_{n-1} y_n M}{R}} ds_n ds_{n-1} + \int_{\frac{y}{M}}^1 \int_{\frac{s_{n-1} y_n M}{R}}^{\frac{s_{n-1}^2 M}{R}} ds_n ds_{n-1} \right] \\
&= 1 - \sum_y p(y) \left[\frac{M}{3R} - \frac{y}{2R} - \frac{y^3}{3M^2 R} \right] \\
\mathcal{L}_{MM} &\leq \sum_y p(y) \left[\frac{M}{3R} - \frac{y}{2R} - \frac{y^3}{3M^2 R} \right]
\end{aligned}
$$

This concludes the proof. □

In the next section, we show that this model not only allows designers to quantify the leakage of their vanilla implementations but also to incorporate different countermeasures and quantify the security guarantees provided by them.

4.5 Countermeasures

A number of countermeasures have been developed to prevent information leakage in cryptosystems. We focus on the two most popular countermeasures: a) exponent blinding and b) caching. Exponent blinding thwarts timing attacks by adding a different random value to the exponent for each exponentiation. In contrast, caching thwarts such attacks by pre-computing the output of certain multiplicand pairs; i.e. maintaining multiplication table. The algorithm does a look-up for all such pairs and therefore, reduces the number of extra reductions. Despite the mainstream belief of the strength of these countermeasures, not many quantitative guarantees are available in the literature. In this section, we quantify the security guarantees of both countermeasures and study their performance trade-offs under resource constraints.

4.5.1 Exponent Blinding

In his seminal paper on timing attacks in cryptosystems [37], Paul Kocher also proposed a countermeasure against such attacks, named exponent blinding. In this countermeasure, the exponent, d, is added with a random multiple of the Euler's totient function, $\phi(M)$ of the modulus M. As opposed to decrypting a ciphertext y by computing $y^d (mod\ M)$, two exponentiations are performed with random exponents. First, $y^{d+r\phi(M)} (mod\ M)$ is computed, followed by $y^{r\phi(M)} (mod\ M)$. The results of these computations are divided to obtain $y^d (mod\ M)$. The salt,

r, is chosen randomly for each ciphertext. As each of these operations is performed with random exponents selected fresh for each ciphertext, original timing attacks are hindered. While traditionally, it was believed that exponent blinding completely defeats all timing attacks, recent works have shown that timing attacks are still possible under this countermeasure albeit with significantly higher number of measurements [2, 59].

We study the efficacy of exponent blinding by accommodating it in the leakage model from previous section. To emulate the effect of a random salt, we assume that along with the user's input bit b_i, a random bit r_i is generated which remains unknown to the attacker. Two Montgomery Multiplication operations are performed for each ciphertext y_i. First multiplication is performed using $b_i \oplus r_i$ as the exponent bit and emulates $y^{d+r\phi(M)}(mod\ M)$ operation. This operation reveals a binary value w_i^1 to the attacker, where $w_i^1 = 1$ if an extra reduction is required and 0 otherwise. Second multiplication is performed using r_i as the exponent bit and emulates $y^{r\phi(M)}(mod\ M)$. This operation reveals another binary value w_i^2 to the attacker, where $w_i^2 = 1$ if an extra reduction is required and 0 otherwise. Given, his knowledge of y_i and $w_i^{(1,2)}$, the goal of the attacker is to learn the user's input bit b_i. The leakage of the system under this countermeasure is measured as:

$$\mathcal{L}_{blind} := \lim_{n\to\infty} 1 - \frac{H(B^n||Y^n, W^{(1,2)n})}{H(B^n)}$$

The computation of $\mathcal{L}_{blind}$ can be performed similarly to the computation of $\mathcal{L}_{MM}$ because the probability of observing an extra reduction in each multiplication is independent of the other. b_i is only estimated correctly if both $r_i \oplus b_i$ and r_i are estimated correctly. Therefore, conditional entropy of B_i given Y_i and $W_i^{(1,2)}$ is only dependent on the conditional entropy of $r_i \oplus b_i$ and r_i, which are identical. Specifically,

Theorem 14. *The upper-bound on the leakage of the Montgomery Multiplication routine for a non-adaptive attack strategy, $p(y)$, under exponent blinding can be computed as*

$$\mathcal{L}_{blind} \leq \sum_y p(y) \left(\frac{M}{3R} - \frac{y}{2R} - \frac{y^3}{3M^2R} \right)^2$$

Proof. In line with the proof of Theorem 13, we have

$$H(B_i|Y_i, W_i^{(1,2)}) \geq H(B_i|Y_i, W_i^{1,2}, S_i^{(1,2)}, S_{i-1}^{(1,2)})$$

Now, b_i is computed with certainty if both $b_i \oplus r_i$ and r_i are computed with certainty. In any other case, $H(B_i|\cdots) = 1$. Therefore, we only need to compute the probability that both exponent bits are learned with certainty. Since the stochastic process for either multiplications is i.i.d. given y_i,

$$P(b_i \text{ learned with certainty}) = P^2(r_i \text{learned with certainty})$$

Given y_i, $P(r_i$ is learned with certainty$)$ is identical to $P(b_i$ is learned with certainty$)$ in the proof of Theorem 13. This probability is computed as.

$$P(r_i \text{ is learned with certainty}) = \frac{M}{3R} - \frac{y}{2R} - \frac{y^3}{3M^2R}$$

Combining these results we have,

$$H(B_i|Y_i, W_i^{1,2}, S_{i,i-1}^{(1,2)}) = 1 - \sum_y p(y) \left[\frac{M}{3R} - \frac{y}{2R} - \frac{y^3}{3M^2R}\right]^2$$

Consequently,

$$\mathcal{L}_{blind} = \sum_y p(y) \left[\frac{M}{3R} - \frac{y}{2R} - \frac{y^3}{3M^2R}\right]^2$$

□

Since $\left(\frac{M}{3R} - \frac{y}{2R} - \frac{y^3}{3M^2R}\right) \leq 1$, it is clear that $\mathcal{L}_{blind} \leq \mathcal{L}_{MM}$. This establishes that blinding does reduce the side-channel information leakage but does not prohibit it entirely. This finding is consistent with the existence of attacks in the presence of blinding.

4.5.2 Caching

Caching thwarts timing attacks by pre-computing the product for certain pairs of multiplicands and caching it in memory. When any such pair is encountered during a Montgomery Multiplication operation, the algorithm looks-up in the table and retrieves the output. This is an $\mathcal{O}(1)$ operation and does not contribute towards the total computation time. It is assumed here that the attacker does not know the contents of the pre-computed multiplication table and cannot select specific ciphertexts to avoid look-ups in the table. We compute the security guarantees provided by this countermeasure under our leakage model and study the performance trade-offs with the amount of memory dedicated to caching.

To accommodate this countermeasure in our leakage model, we assume that whenever a pre-computed multiplication is required, the attacker does not know if a reduction is required or not. In such cases, the output of the side channel, w_i, is neither 0 or 1 but an erasure which is denoted by e. The probability that the attacker observes an erasure depends on the size of the multiplication table. If the scheme has a memory budget which allows the system to store Θ number of multiplicand pairs, the given two randomly chosen inputs the probability of the attacker observing an erasure equals $\frac{\Theta}{M^2}$. We define the leakage of the modular multiplication routine under this countermeasure as

$$\mathcal{L}_{cache} = \lim_{n\to\infty} 1 - \frac{H(B^n||Y^n, W^n)}{H(B^n)}$$

where $w \in \{0, 1, e\}$.

Theorem 15. *The upper-bound on the leakage of the Montgomery Multiplication routine for a non-adaptive attack strategy, $p(y)$, when caching is employed can be computed as,*

$$\mathcal{L}_{cache} \leq \left(1 - \frac{\Theta}{M^2}\right) \mathcal{L}_{MM}.$$

Proof. The leakage of this implementation is the same as the vanilla Montgomery Multiplication routine if the multiplicands are not cached. When the multiplicands are cached, the attacker

observes an erasure and does not gain any information about the user's input bit. Averaging both cases, we get $\mathcal{L}_{cache} = (1 - \alpha)\mathcal{L}_{MM}$. □

An important implication of Theorem 14 and Theorem 15 is that the reduction in leakage achieved by both countermeasures is inherently different in nature. Caching reduces the leakage by a constant multiplier that depends on the memory budget. In contrast, blinding reduces the leakage of the system by an order, $\mathcal{L}_{blind} \approx \mathcal{L}^2$. Therefore, the choice of countermeasure must depend on the leakage of the vanilla implementation alongside other factors such as the memory budget. If the leakage of the vanilla implementation is already low; i.e., $\mathcal{L} \to 0$, using blinding reduces it further. For this case, caching requires very high memory budget to reduce the leakage at the same level as blinding. In contrast, if the leakage of the vanilla implementation is high; i.e., $\mathcal{L} \to 1$, reduction in leakage achieved by blinding is not significant and therefore, caching is a preferred countermeasure for such scenarios.

4.6 Conclusions

We employed stochastic models developed for the Montgomery Multiplication routine to analyze reliability rate of an attacker who attempts to learn the underlying secret modulus. Our results show that the reliability rate is non-zero and inversely proportional to the Montgomery Multiplication reduction parameter, R. Additionally, we developed a new side-channel model for Montgomery Multiplication that allows us to measure the asymptotic leakage of the side channel. Under this model we are able to quantify the security provided by well-known countermeasures: exponent blinding and caching. We show that the reduction in leakage achieved by both countermeasures are fundamentally different. While exponent blinding reduces leakage by an order, caching reduces the leakage by a constant factor that depends on the memory budget of the countermeasure. This led to identify the conditions under which one countermeasure outperforms the other.

Chapter 5

Conclusions and Future Work

This thesis has shown that quantification of information leakage through side-channel attacks helps find parameters that leak least information, optimal attack strategies that increase the impact of an attack several folds, and finally, countermeasures that are practical and provably-secure. The key to this quantitative analysis is the modeling of a side channel and choice of leakage metrics. We have developed a model that treats a side channel as a two-input-single-output system where the statistical relationship between inputs and outputs defines the side-channel. We showed how this model can be used to define precisely a variety of side-channels attacks, such as private communication detection, privacy attacks against packet schedulers, and timing attacks against cryptosystems.

In this thesis, we analyze three types of metrics: *capacity*, *reliability rate*, and *leakage*. We show that capacity is an ill-suited metric as it cannot be associated with a negligible probability of error of an attacker. Reliability rate is computed in terms of the error-exponent of an attacker is estimating user's secret, whereas leakage is measured in terms of the mutual-information rate between side-channel output and inputs. Reliability rate quantifies the accuracy with which the attacker learns the secret whereas leakage quantifies the amount of information leaked per-side channel input. These two metrics are distinct; that is, security of a system under leakage criteria does not imply its security under reliability rate criteria, and vice versa.

With a useful model and metrics in place, we analyzed three different side-channel attacks. We developed a new stochastic definition of private communication detection between two parties and used system models available in the literature for packet schedulers and modular multiplication based cryptosystem. Under these models, we computed the reliability rate of an attacker in estimating private communication relationships and secret RSA primes. We were able to study the effect of different parameters choices on these reliability rates, such as probing and communication rates in PCD and the Montgomery reduction parameter in modular multiplication. We showed that while reliability rate measures the accuracy of an attacker in estimating secret information, its analysis is specific to an attack setup and cannot be generalized. For generalized analyses we need the leakage metric.

Using our leakage metric, we were able to compute the rate as which activity-logs leak information about call-records in Private Communication Detection, the rate at which optimal strategies leak information about packet arrivals at a scheduler, and the rate at which extra reductions in the Montgomery Multiplication routine leak information about secret exponent bits. For packet schedulers, we show that an non-adaptive adversary is able to find optimal strategies that cause 1000% more leakage than previously reported in literature. Furthermore,

we showed that adaptive strategies lead to even higher leakage and therefore, must be considered in side-channel analyses. For the Montgomery Multiplication routine, we introduced a unique side-channel model which allowed computation of asymptotic leakage. Due to high memory of this channel, we were not able to compute or identify optimal attack strategies but we developed lower and upper-bounds on leakage of non-adaptive strategies.

Analysis of information leakage also allowed us to develop strong countermeasures, such as resource-randomization against PCD, and analyze the efficacy of known countermeasures under practical conditions. We proposed addition of noise in the PCD side-channel by randomizing the use of allocated resources and showed that this countermeasure can be used to prevent any leakage from the side channel. For timing attacks against cryptosystems, we were able to differentiate between two well-known countermeasures, exponent blinding and caching, and explain the differences between their security behavior. Lastly, we were able to identify conditions under which one outperforms the other.

Several interesting questions are left for future research.

Q1: First question pertains to studying the relationship between leakage and reliability rate metrics. In particular, it seems intuitive that security of a system under leakage criteria implies that the attack cannot distinguish between user's side-channel inputs. This is likely to shed new light on the reliability rate that can be achieved by an attacker. It would be of interest to study the side-channel conditions and usage mappings under which such relationships can be established.

Q2: For the timing side channel in modular exponentiation-based cryptographic algorithms, it is important to associate the leakage of the Montgomery Multiplication routine with the leakage of a general cryptographic algorithm. Intuitively, the amount of information provided due to the knowledge of individual extra reductions is higher than when the attacker only knows their sum; i.e. total computation times. However, the scaling of leakage with the key-size is still not established for a general attack strategy. Finally, the analysis of leakage of the Montgomery Multiplication routine for all adaptive strategies is required to compute its worst-case leakage.

While the side-channel model proposed in this thesis fits a number of real-world scenarios, certain extensions of it are of interest to the research community.

E1: Consider the scenario in which the attacker manages to insert a Trojan horse in the system or the user's device, which leaks information to the attacker through a parallel, low-capacity covert channel mechanism. In such case, the definition of *joint* side-covert side channel capacity becomes necessary to compute the leakage of the system under this setup. Intuitively, the joint leakage would be higher than the case without the covert channel and method is necessary to quantify it.

E2: Another extension relates to the design and analysis of *adaptive countermeasures*. The countermeasures presented in the literature and this thesis are typically deterministic and operate assuming the worst-case behavior from the attacker. An interesting research question is to design and compute the efficacy of *adaptive countermeasures* which observe previous behavior from the attacker to adjust countermeasure parameters in a way. This has the potential to thwart most/all attacks with minimal performance penalties. In the most general case, the attacker and the defender can both be adaptive. Game-theoretic formulations may be required to analyze such systems.

Appendix A

Anonymity leakage in communication systems

A.1 Probability of call-records given activity-logs

If Alice and Bob communicate with each other, the probability of observing a joint activity-log (al^n) is equal to the sum of probabilities of all call records, cr^n, under the Markov model shown in Figure 2.4 which lead to the same al^n. Let, $\mathcal{T}(al^n) \equiv \{cr^n, \text{ s.t. } cr^n \text{ map to } al^n\}$. Then,

$$P(al^n|H_1) \equiv \overline{p}^r(al^n) = \sum_{cr^n \in \mathcal{T}(al^n)} \overline{p}^r(cr^n).$$

From Figure 2.5, we can deduce that the only communication states that lead to confusion in activity status are 11 and $\overline{11}$. $P(al^n|H_1)$ depends only on the number of transitions of the type $x \rightarrow y$ ($O(x \rightarrow y)$), and the number of sub-sequences of the type $x - (11)^k - y$: ($O(x - (11)^k - y)$), for $x, y \in \{00, 01, 11\}$ and $k \in \{1, \cdots, n-2\}$. Therefore, $P(al^n|H_1)$ can be written using these parameters as:

$$\overline{p}(al^n) = \pi_0 \prod_{x,y,k} \overline{p}_{x\rightarrow y}^{O(x\rightarrow y)} \overline{p}(x - (11/\overline{11})^k - y)^{O(x-(11)^k-y)}$$

Here, we can ignore the initiating and terminating sub-sequences, $11^k - y$ and $x - 11^k$, as they can only occur once in the sequence and therefore, their asymptotic contribution is zero. Similarly, an all 11^n activity-log sequence can be ignored. The probability of the sub-sequence $x - (11)^k - y$

$$\overline{p}(x - (11/\overline{11})^k - y) = \pi_x \begin{bmatrix} \overline{p}^r_{x\rightarrow 11} & \overline{p}^r_{x\rightarrow \overline{11}} \end{bmatrix} \begin{bmatrix} \overline{p}^r_{11\rightarrow 11} & \overline{p}^r_{11\rightarrow \overline{11}} \\ \overline{p}^r_{\overline{11}\rightarrow 11} & \overline{p}^r_{\overline{11}\rightarrow \overline{11}} \end{bmatrix}^k \begin{bmatrix} \overline{p}^r_{11\rightarrow y} \\ \overline{p}^r_{\overline{11}\rightarrow y} \end{bmatrix}$$

Finally, the KL divergence $D(P(al^n|H_0)||P(al^n|H_1))$,

$$= -\sum_{al^n} p(al^n) \log \frac{\overline{p}(al^n)}{p(al^n)}$$

$$
\begin{aligned}
&= -\sum_{al^n}\sum_{x,y} p(al^n)O(x \to y)\log\frac{\overline{p}_{x\to y}}{p_{x\to y}} \\
&- \sum_{al^n}\sum_{x,y,k} p(al^n)O(x-(11)^k-y)\log\frac{\overline{p}_{x-(11/\overline{11}^k)y}}{p_{x-(11)^k-y}}
\end{aligned}
$$

Finally,

$$\lim_{n\to\infty}\frac{\sum_{al^n} p(al^n)O(x\to y)}{n} = \pi_x p_{x\to y}$$

and

$$\lim_{n\to\infty}\frac{\sum_{al^n} p(al^n)O(x-(11)^k-y)}{n} = p_{x-(11)^k-y}.$$

Appendix B

Proof of Theorem 9

The analysis is restricted to time-invariant probability distributions, $p(a_n = a|\bar{q}_{n-1} = q) = p(a|\bar{q})$.

Proof. The proof is performed in four parts: 1) We show that it suffices for the attacker to use partial history $\bar{q}_n$ to determine a_{n+1}, 2) We identify the conditions required for leakage to have an asymptotic limit, 3) We show that this conditions are satisfied for time-invariant distribution, and 4) We show that limiting distribution of the state $\bar{q}_n$ can be computed as the stationary distribution of a Markov chain. We start with the proof of 1).

1) As can be seen from the functional-dependence graphs in Figure 3.7a and 3.7b, given the queuing delay of a packet, the queuing delays of all future packets depend only of the delay of the current packet and inter-arrivals of future packets. Thus, the queuing delay and inter-arrival times of past packets can be ignored. This implies that the system's behavior can be represented as a MDP. At the same time, rewards in this MDP depend only of the previous state, $\bar{q}_n$ and the action a_{n+1}. We can again apply dominance of Markov policies to show that the optimal strategies in this MDP is also Markovian; i.e., of the form $p(a_{n+1}|\bar{q}_n)$.

Next, we determine the conditions for $\lim_{n\to\infty} H(X^n||A^n)$ and $\lim_{n\to\infty} H(X^n||A^n, D^n)$ to exist and be computable under Césaro's mean theorem.

2) First,

$$\begin{aligned}
\lim_{n\to\infty} H(X^n||A^n) &= \lim_{n\to\infty} \sum_n H(X_n|A_n) \\
&= \lim_{n\to\infty} \sum_{a_n} p(a_n) H(X_n|a_n) \\
&= \lim_{n\to\infty} \sum_{\bar{q}_{n-1}, a_n} p(\bar{q}_{n-1}) p(a_n|\bar{q}_{n-1}) H(X_n|a_n) \\
&= \sum_{\bar{q}_{n-1}, a_n} \Big(\lim_{n\to\infty} p(\bar{q}_{n-1}) \Big) p(a_n|\bar{q}_{n-1}) H_B(\lambda_1, a_n)
\end{aligned}$$

This is because, $p(a_n|\bar{q}_{n-1})$ is assumed to be time-invariant and $H_B(\lambda_1, a_n)$ depends only on a_n. Similarly,

$$\lim_{n\to\infty} H(X^n||A^n, D^n) = \lim_{n\to\infty} \sum_n H(X_n|A_n, D_n, D_{n-1})$$

$$
\begin{aligned}
&= \lim_{n\to\infty} \sum_{a_n, d_{n-1}} p(a_n, d_{n-1}) P_{EQ}(a_n, d_{n-1}) H(X_{a_n, d_{n-1}})) \\
&= \lim_{n\to\infty} \sum_{\bar{q}_{n-1}, a_n, d_{n-1}} p(\bar{q}_{n-1}) p(D_{n-1}|\bar{q}_{n-1}) p(a_n|\bar{q}_{n_1}) \\
&\quad P_{EQ}(a_n, d_{n-1}) H(X_{a_n, d_{n-1}}) \\
&= \sum_{\bar{q}_{n-1}, a_n, d_{n-1}} \left(\lim_{n\to\infty} p(\bar{q}_{n-1})\right) \left(\lim_{n\to\infty} p(d_{n-1}|\bar{q}_{n-1})\right) p(a_n|\bar{q}_{n-1}) \\
&\quad P_{EQ}(a_n, d_{n-1}) H(X_{a_n, d_{n-1}})
\end{aligned}
$$

Clearly, the limit $\mathcal{L}^c$ exists only if the required limiting distributions exist. Now we show that these limiting distributions exist for strategies under consideration.

3) We show that $\lim_{n\to\infty} p(d_{n-1}|\bar{q}_{n-1})$ exists. Given $\bar{q}_{n-1} \equiv s_{n-1}, d_{n-1-s_{n-1}}, [a_j]_{n-1-s_{n-1}}^{n-1}$, d_{n-1} can be simply determined as

$$
d_{n-1} = \left[d_{n-1-s_{n-1}} + \sum_{j=n-s_{n-1}}^{n-1} (x_j + 1 - a_j) \right]^+ .
$$

Given, a_j, x_j is statistically independent of other parameters. Therefore, for $a_{n-1} \neq 0$,

$$
p(d_{n-1}|\bar{q}_{n-1}) = \binom{\sum_j a_j}{x^*} (1-\lambda_1)^{\sum_j a_j - x^*} \lambda_1^{x^*},
$$

where $x^* = d_{n-1} - d_{n-1-s_{n-1}} + \sum_j (a_j - 1)$.

For, $d_{n-1} = 1$,

$$
p(d_{n-1} = 1|\bar{q}_{n-1}) = \sum_{d=1}^{\infty} p(d_{n-1} = d|\bar{q}_{n-1})
$$

Clearly, d_{n-1} depends only on the value of $\bar{q}_{n-1}$ and is independent of n. Therefore, the limiting distribution is given trivially by the above equation.

4) Finally, we show that the generation of the sequence $\bar{q}_1, \bar{q}_2, \cdots$ is a first-order, irreducible, and a-periodic Markov-chain. Therefore, the limiting distribution $\lim_{n\to\infty} p(\bar{q}_n)$ exists and equals the stationary distribution of the Markov chain. We start by showing that conditioned on $\bar{q}_n$, $\bar{q}_{n+1}$ is independent of $\bar{q}_{n-1}, \bar{q}_{n-2}, \cdots$. The inter-arrival time a_{n+1} is determined by the attack strategy on the basis of $\bar{q}_n$ and therefore, is entirely dependent on it. s_{n+1} is completely determined by the following relation.

$$
d_{n-s_n} + \sum_{j=n+1-s_n}^{n+1-s_{n+1}} (x_j - a_j + 1) \leq a_{n+1} \leq d_{n-s_n} + \sum_{j=n+1-s_n}^{n+1-s_{n+1}} (x_j - a_j + 1) + x_{n+2-s_{n+1}}
$$

Similarly, given s_{n+1}, $a_{n+1-s_{n+1}}$ is determined by the following relation

$$
d_{n+1-s_{n+1}} = \left[d_{n-s_n} + \sum_{j=n+1-s_n}^{n+1-s_{n+1}} (x_j - a_j + 1) \right]^+
$$

Clearly, all the terms in the above relation are either constituents of $\bar{q}_n$ or are determined by it. This shows that the stochastic process $\bar{q}_1, \bar{q}_2, \cdots$ is a first-order Markov chain. The same can be confirmed by analyzing the fd-graph between the states for this strategy as shown in Fig.3.7a and 3.7b where it can be readily seen that q_n $d-$separates q_{n+1} from $q_{n-1}, \cdots$. The state transition probabilities for the Markov chain are given as

$$\begin{aligned} p(\bar{q}_{n+1}|\bar{q}_n) &= p(s_{n+1}, d_{n+1-s_{n+1}}, [a_j]_{n+1-s_{n+1}^{n+1}}|\bar{q}_n) \\ &= p(a_{n+1}|\bar{q}_n)p(s_{n+1}, d_{n+1-s_{n+1}}|q_n, a_{n+1}) \end{aligned}$$

For $d_{n+1-s_{n+1}} \leq a_{n+1}$ and $x_{n+2-s_{n+1}} \geq a_{n+1} + 1 - d_{n+1-s_{n+1}}$, $p(s_{n+1}, d_{n+1-s_{n+1}}|q_n, a_{n+1})$

$$= \binom{\sum_{j=n+1-s_n}^{n+1} a_j}{x^*} (1-\lambda_1)^{\sum_{j=n+1-s_n}^{n+1} a_j - x^*} \lambda_1^{x^*}$$

where, $x^* = d_{n+1-s_{n+1}} - d_{n-1-s_{n-1}} + \sum_j (a_j - 1)$

For all other pairs (q_{n+1}, q_n), $p(q_{n+1}|q_n) = 0$. It can be seen easily from the transition probabilities that the Markov chain is a-periodic and is a single communicating class, and therefore irreducible. Therefore, the limiting distribution exists and equals the stationary distribution of the Markov chain. □

Appendix C

Analysis of Modular Multiplication-based Cryptographic Algorithms

C.1 Proof of Lemma 2

Proof. Since $temp$ behaves like a random variable equi-distributed on Z_M, the behavior of $\frac{temp}{M}$ is like a random variable uniformly distributed in the range $(0, 1)$. Similarly, the second summand $\frac{y \times temp \times M^*(mod\ R)}{R}$ also behaves as random variable uniformly distributed in the range $(0, 1)$. These variables are independent of each other.

Let $U := \frac{temp}{M}$ and $V := \frac{y \times temp \times M^*(mod\ R)}{R}$. Then, an extra reduction in the computation of $y \times temp(mod\ M)$ is performed if $\frac{y(mod\ M)}{R}U + V \geq 1$. This probability of this event can be computed as follows.

$$\begin{aligned} Pr\left[\frac{y(mod\ M)}{R}U + V \geq 1\right] &= \int_0^1 \int_{1-\frac{y(mod\ M)}{R}u}^1 dv\ du \\ &= \int_0^1 \frac{y(mod\ M)}{R}u\ du \\ &= \frac{y(mod\ M)}{2R} \end{aligned}$$

Similarly, an extra reduction in the computation of $temp^2(mod\ M)$ is performed is $\frac{M}{R}U^2 + V \geq 1$. The probability of an extra reduction in this case can be computed as

$$\begin{aligned} Pr\left[\frac{M}{R}U^2 + V \geq 1\right] &= \int_0^1 \int_{1-\frac{M}{R}u^2}^1 dv\ du \\ &= \int_0^1 \frac{M}{R}u^2\ du \\ &= \frac{M}{3R} \end{aligned}$$

□

C.2 Timing Behavior for Modular Exponentiation with Unknown Modulus

We assume that a regular multiplication requires c seconds in the absence of an extra reduction. Extra reduction adds c_{ER} seconds to a multiplication operation. Let, $|d|$ and d_1 be the total number of bits and total number of ones in the binary-representation of the exponent d. In the computation of $y^d(mod\ M)$, the total number of $temp^2(mod\ M)$ operations equal $|d|$ and total number of $y \times temp(mod\ M)$ operations equal d_1. For cryptographic exponents, $d_1 \approx \frac{|d|}{2}$. The probability of an extra reduction is different for each type of multiplication. On average, the number of extra reductions equal $|d|\frac{m}{3R} + d_1\frac{y(mod\ M)}{2R}$.

We now focus on the sequential Montgomery Multiplications that are performed in modular exponentiation using the *square-and-multiply* algorithm. Let, $S_i := \frac{temp_i}{R}$ before the $i + 1^{th}$ multiplication and V_{i+1} represent a random variable uniformly-distributed on $(0, 1)$. We have,

$$S_{i+1} = \begin{cases} \frac{M}{R}S_i^2 + V_{i+1} & \text{for } MM(temp_i, temp_i) \\ \frac{y}{M}\frac{M}{R}S_i + V_{i+1} & \text{for } MM(temp_i, y_i) \end{cases}$$

Similarly, let $W_{i+1} \in \{0, 1\}$ represent a binary random variable that represents whether $i+1^{th}$ multiplication required an extra reduction. S_i's behave as independent random variables, uniformly-distributed over $(0, 1)$. From Lemma 2, the random variable W_i is defined as:

$$W_i = \begin{cases} 1_{S_i < \frac{M}{R}S_{i-1}^2} & \text{for } MM(temp_i, temp_i) \\ 1_{\S_i < \frac{y}{M}\frac{M}{R}S_{i-1}} & \text{for } MM(temp_i, y) \end{cases}$$

Then, the total time $T(y)$ to compute $y^d(mod\ M)$ can be expressed as

$$T(y) = c|d| + c_{ER}\sum_{i=1}^{|d|+d_1} W_i$$

We have, the expected value of W_i

$$\mathbf{E}[W_i] = \begin{cases} \frac{M}{3R} & \text{for } MM(temp, temp) \\ \frac{y(mod\ m)}{2R} & \text{for } MM(temp, y) \end{cases}$$

and the variance of W_i

$$Var[W_i] = \begin{cases} \frac{M}{3R} - \left(\frac{M}{3R}\right)^2 & \text{for } MM(temp, temp) \\ \frac{y}{2R} - \left(\frac{y}{2R}\right)^2 & \text{for } MM(temp, y) \end{cases}$$

However, the sequence of random variables $\{W_i\}$ are neither independent nor identically distributed. The co-variance between W_i, W_{i+1} can be computed in the following way for three different cases:

Case I: ($W_i \Leftarrow MM(temp_i, temp_i)$ and $W_{i+1} \Leftarrow MM(temp_{i+1}, y)$)

$$\begin{aligned} Cov_{SM} &= \mathbf{E}(W_iW_{i+1}) - \mathbf{E}(W_i)\mathbf{E}(W_{i+1}) \\ &= \mathbf{E}(W_iW_{i+1}) - \frac{M}{3R}\frac{y}{2R} \end{aligned}$$

$$
\begin{aligned}
&= Pr[W_i = 1 \cap W_{i+1} = 1] - \frac{M}{3R}\frac{y}{2R} \\
&= \int_0^1 \int_0^{\frac{M}{R}s_{i-1}^2} \int_0^{\frac{y}{R}s_i} ds_{i+1} ds_i ds_{i-1} - \frac{M}{3R}\frac{y}{2R} \\
&= \frac{1}{10}\frac{M^2}{R^2}\frac{y}{R} - \frac{M}{3R}\frac{M}{2R}
\end{aligned}
$$

Case II:($W_i \Leftarrow MM(temp, y)$ and $W_{i+1} \Leftarrow MM(temp, temp)$)

$$
\begin{aligned}
Cov_{MS} &= \mathbf{E}(W_i W_{i+1}) - \mathbf{E}(W_i)\mathbf{E}(W_{i+1}) \\
&= \mathbf{E}(W_i W_{i+1}) - \frac{y}{2R}\frac{M}{3R} \\
&= Pr[W_i = 1 \cap W_{i+1} = 1] - \frac{y}{2R}\frac{M}{3R} \\
&= \int_0^1 \int_0^{\frac{y}{R}s_{i-1}} \int_0^{\frac{M}{R}s_i^2} ds_{i+1} ds_i ds_{i-1} - \frac{y}{2R}\frac{M}{3R} \\
&= \frac{1}{12}\frac{y^3}{R^3}\frac{M}{R} - \frac{M}{3R}\frac{M}{2R}
\end{aligned}
$$

Case III:($W_i \Leftarrow MM(temp, temp)$ and $W_{i+1} \Leftarrow MM(temp, temp)$)

$$
\begin{aligned}
Cov_{SS} &= \mathbf{E}(W_i W_{i+1}) - \mathbf{E}(W_i)\mathbf{E}(W_{i+1}) \\
&= \mathbf{E}(W_i W_{i+1}) - \frac{M}{3R}\frac{M}{3R} \\
&= Pr[W_i = 1 \cap W_{i+1} = 1] - \frac{M}{3R}\frac{M}{3R} \\
&= \int_0^1 \int_0^{\frac{M}{R}s_{i-1}^2} \int_0^{\frac{M}{R}s_i^2} ds_{i+1} ds_i ds_{i-1} - \frac{M}{3R}\frac{M}{2R} \\
&= \frac{1}{21}\frac{M^4}{R^4} - \frac{M}{3R}\frac{M}{2R}
\end{aligned}
$$

It is easy to see that $Cov(W_i, W_j) = 0$, iff $|i - j| > 1$. Thus, we can invoke the central limit theorem for loosely-independent random variables to model the sum $\sum_{i=1}^{|d|+d_1} W_i$.

C.3 Proof of Lemma 3

For a given probability distribution Q in y,

$$
R(i, Q) = \frac{1}{2\sigma^2} \min_{j \neq i} \sum_y q(y)[(\mu_{y,M_i} - \mu_{y,M_j})^2].
$$

Substituting μ_{y,M_i} and μ_{y,M_j}, we get

$$(\mu_{y,M_i} - \mu_{y,M_j})^2 = c_{ER}^2 \left[\frac{|d|(M_i - M_j)}{3R} + \frac{d_1(y(mod\ M_i) - y(mod\ M_j))}{2R} \right]$$

Define, $\Delta M := M_i - M_j$ and $\Delta y(mod\ M) = y(mod\ M_i) - y(mod\ M_j)$.

$$\begin{aligned} R(i,Q) &= \frac{c_{ER}^2|d|^2}{2\sigma^2 R^2} \mathbb{E}_y \left[\frac{\Delta M}{3} + \frac{\Delta y(mod\ M)}{4} \right]^2 \\ &= \frac{c_{ER}^2|d|^2}{2\sigma^2 R^2} \left[\frac{\Delta M^2}{9} + \frac{\Delta M \mathbb{E}_y(\Delta y(mod\ M))}{6} + \frac{\mathbb{E}_y^2(\Delta y(mod\ M))}{16} \right] \end{aligned}$$

The value of $\Delta y(mod\ M)$, depends on whether $M_j < M_i$ and $M_i < M_j$. We focus only on values $0 < y < \max M$. This restriction ensures that $y < 2M_j$ since M_i and M_j have the same number of bits.

Case I: $(M_j < M_i)$

$$\Delta y(mod\ M) = \begin{cases} 0 & \text{if } y \leq M_j - 1 \\ M_j & \text{if } M_j \leq y \leq M_i - 1 \\ -\Delta M & \text{if } M_i \leq y \end{cases}$$

$$\begin{aligned} \mathbb{E}_y(\Delta y(mod\ M)) &= M_j\{F_Q(M_i - 1) - F_Q(M_j)\} - \Delta M\{1 - F_Q(M_i)\} \\ \mathbb{E}_y^2(\Delta y(mod\ M)) &= M_j^2\{F_Q(M_i - 1) - F_Q(M_j)\} + \Delta^2 M\{1 - F_Q(M_i)\} \end{aligned}$$

Substituting these values in (4), we get

$$\begin{aligned} R(i,Q) &= \frac{c_{ER}^2|d|^2}{2\sigma^2 R^2} \min_{M_j < M_i} \left\{ \Delta M^2 \left[\frac{1}{9} - \frac{5}{48}[1 - F_y(M_i)] \right] \right. \\ &\quad \left. + \left[\frac{M_j \Delta M}{6} + \frac{M_j^2}{16} \right] [F_y(M_i - 1) - F_y(M_j)] \right\} \end{aligned}$$

As $M_j \to M_i$; $\Delta M \to 0$, $F_y(M_i - 1) - F_y(M_j) \to 0$. The only term that behaves differently is,

$$\frac{M_j \Delta M}{6} + \frac{M_j^2}{16} = \frac{M_j(8M_i - 5M_j)}{48}$$

Since $\frac{M_i}{2} \leq M_j < M_i$ and $M_j(8M_i - 5M_j)$ is a parabola that achieves its maximum value at $M_j = \frac{4}{5}M_i$. This means that (5) has the same value for $\frac{3}{5}M_i \leq M_j \leq \frac{4}{5}M_i$ and $\frac{4}{5}M_i \leq M_j < M_i$. $R(i,Q)$ is lower for the later range. Therefore, the comparison needs to be for $\frac{1}{2}M_i \leq M_j \leq \frac{3}{5}M_i$ and $\frac{4}{5}M_i \leq M_j < M_i$. Precisely, it is between the values $M_j \approx \frac{1}{2}M_i$ or $M_j = M_{i-1}$. Let,

$$\begin{aligned} R_l(i,Q) &:= \frac{c_{ER}^2|d|^2}{2\sigma^2 R^2} \left\{ \Delta_{i-1}^2 \left[\frac{1}{9} - \frac{5}{48}(1 - F_y(M_i)) \right] \right. \\ &\quad \left. + \left[\frac{M_{i-1}\Delta_{i-1}}{6} + \frac{M_{i-1}^2}{16} \right] [F_y(M_i - 1) - F_y(M_{i-1})] \right\} \end{aligned}$$

Next, we perform the analysis of $M_i < M_j$.
Case II: $(M_i < M_j)$

$$\Delta y(mod\ M) = \begin{cases} 0 & \text{if } y \leq M_i - 1 \\ -M_i & \text{if } M_i \leq y \leq M_j - 1 \\ -\Delta M & \text{if } M_j \leq y \end{cases}$$

$$\begin{aligned} \mathbb{E}_y(\Delta y(mod\ M)) &= -M_i\{F_Q(M_j - 1) - F_Q(M_i)\} - \Delta M\{1 - F_Q(M_j)\} \\ \mathbb{E}_y^2(\Delta y(mod\ M)) &= M_i^2\{F_Q(M_j - 1) - F_Q(M_i)\} + \Delta^2 M\{1 - F_Q(M_j)\} \end{aligned}$$

Substituting these values in (4), we get

$$\begin{aligned} R(i,Q) &= \frac{c_{ER}^2|d|^2}{2\sigma^2 R^2} \min_{M_i < M_j} \left\{ \Delta M^2 \left[\frac{1}{9} - \frac{5}{48}[1 - F_y(M_j)] \right] \right. \\ &\quad \left. + \left[\frac{-M_i \Delta M}{6} + \frac{M_i^2}{16} \right] [F_y(M_j - 1) - F_y(M_i)] \right\} \end{aligned}$$

As $M_i \leftarrow M_j$; $\Delta M \rightarrow 0$, $F_y(M_i - 1) - F_y(M_j) \rightarrow 0$. The only term that behaves differently is,

$$\frac{-M_i \Delta M}{6} + \frac{M_i^2}{16} = \frac{M_i(8M_j - 5M_i)}{48}$$

Since $M_i < M_j \leq 2M_i$ and $M_j(8M_i - 5M_j)$ is a parabola that is an decreasing function of M_j in the specified range, the minimum value is achieved for $M_j = M_{i+1}$. Combining results from the two cases, we can be certain that irrespective of the choice of input distribution Q, the minimum value is achieved for $M_j \in \{M_{i-1}, M_{i+1}\}$. This result reduces the search space for the $\min_{M_j \neq M_i} D()$ to three values, reducing the complexity of search significantly. Let,

$$\begin{aligned} R_u(i,Q) &:= \frac{c_{ER}^2|d|^2}{2\sigma^2 R^2} \left\{ \Delta_i^2 \left[\frac{1}{9} - \frac{5}{48}(1 - F_y(M_{i+1})) \right] \right. \\ &\quad \left. + \left[\frac{M_i \Delta_i}{6} + \frac{M_i^2}{16} \right] [F_y(M_{i+1} - 1) - F_y(M_i)] \right\} \end{aligned}$$

Therefore,

$$R(i,Q) = \min\{R_l(i,Q), R_u(i,Q)\}$$

C.4 Proof of Lemma 4

Replacing Δ_{i-1} and Δ_i with $|d| \log_2 e$ in $R_l(i,Q)$ and $R_u(i,Q)$ respectively, we get

$$\begin{aligned} R_l(i,Q) &= \frac{c_{ER}^2|d|^2}{2\sigma^2 R^2} \left\{ |d|^2 \left[\frac{1}{9} - \frac{5}{48}(1 - F_y(p_i)) \right] + \left[\frac{p_{i-1}|d|}{6} + \frac{p_{i-1}^2}{16} \right] [F_y(p_i - 1) - F_y(p_{i-1})] \right\} \\ R_u(i,Q) &= \frac{c_{ER}^2|d|^2}{2\sigma^2 R^2} \left\{ |d|^2 \left[\frac{1}{9} - \frac{5}{48}(1 - F_y(p_{i+1})) \right] + \left[\frac{p_i|d|}{6} + \frac{p_i^2}{16} \right] [F_y(p_{i+1} - 1) - F_y(p_i)] \right\} \end{aligned}$$

$R_l(i,Q)$ can be maximized without impacting $R_u(i,Q)$ by setting $F_y(p_{i-1}) = 0$. This means that optimal Q should not assign any probability to $y \leq p_{i-1}$. Similarly, $R_u(i,Q)$ can be maximized by setting $F_y(p_{i+1}) = 1$. This implies that optimal Q should assign all probability mass in the range $y \in \{p_{i-1}, p_{i+1}\}$. In that case,

$$\begin{aligned} R_l(i,Q) &= \frac{c_{ER}^2|d|^2}{2\sigma^2R^2}\left\{|d|^2\left[\frac{1}{9} - \frac{5}{48}(1-F_y(p_i))\right] + \left[\frac{p_{i-1}|d|}{6} + \frac{p_{i-1}^2}{16}\right]F_y(p_i)\right\} \\ R_u(i,Q) &= \frac{c_{ER}^2|d|^2}{2\sigma^2R^2}\left\{|d|^2\frac{1}{9} + \left[\frac{p_i|d|}{6} + \frac{p_i^2}{16}\right][1-F_y(p_i)]\right\} \end{aligned}$$

Clearly, $R_l(i,Q)$ is a linearly-increasing function of $F_y(p_i)$ and $R_u(i,Q)$ is a linearly-decreasing function of $F_y(p_i)$. Therefore to compute $\bar{R}(i)$; i.e $\max_Q \min\{R_l(i,Q), R_u(i,Q)\}$ one must compute the value of $F_y(p_i)$ such that $R_l(i,Q) = R_u(i,Q)$.

Denoting $\gamma := 1 - F_y(p_i)$ and $\alpha := \left[\frac{p_i|d|}{6} + \frac{p_i^2}{16}\right] \approx \left[\frac{p_{i-1}|d|}{6} + \frac{p_{i-1}^2}{16}\right]$, $R_l(i,Q) = R_u(i,Q)$ we have,

$$\begin{aligned} -\frac{5|d|^2}{48}\gamma + \alpha(1-\gamma) &= \alpha\gamma \\ \gamma &= \frac{\alpha}{2\alpha + \frac{5|d|^2}{48}} \end{aligned}$$

Substituting γ in $R_u(i,Q)$, we get

$$\bar{R}(i) = \frac{c_{ER}^2|d|^2}{2\sigma^2R^2}\left\{\frac{|d|^2}{9} + \frac{\alpha^2}{2\alpha + \frac{5|d|^2}{48}}\right\}$$

Since $p \gg \log_2(p) = |d|$, we have $2\alpha + \frac{5|d|^2}{48} \approx 2\alpha$, $\alpha \approx \frac{p_i^2}{16}$. This leads to

$$\bar{R}(i) \approx \frac{c_{ER}^2|d|^2p_i^2}{32\sigma^2R^2}.$$

Bibliography

[1] E. Altman. *Constrained Markov Decision Processes*. Stochastic Modeling Series. Taylor & Francis, 1999. ISBN 9780849303821. 3.4.1

[2] S. Bauer. Attacking exponent blinding in rsa without crt. In *Proceedings of the Third International Conference on Constructive Side-Channel Analysis and Secure Design*, COSADE'12, pages 82–88, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-29911-7. doi: 10.1007/978-3-642-29912-4-7. 4.5.1

[3] O. Berthold, H. Federrath, and S. Köpsell. Web mixes: A system for anonymous and unobservable internet access. In *International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, pages 115–129, New York, NY, USA, 2001. Springer-Verlag New York, Inc. ISBN 3-540-41724-9. 1.1.1

[4] O. Berthold, A. Pfitzmann, and R. Standtke. The disadvantages of free mix routes and how to overcome them. In *International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, pages 30–45, New York, NY, USA, 2001. Springer-Verlag New York, Inc. ISBN 3-540-41724-9. 1.2

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787. 3.2.2

[6] D. Brumley and D. Boneh. Remote timing attacks are practical. In *Proceedings of the 12th Conference on USENIX Security Symposium - Volume 12*, SSYM'03, pages 1–1, Berkeley, CA, USA, 2003. USENIX Association. 1, 1.1.2, 1.1.2, 1.2, 3.4, 4

[7] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. Anonymity protocols as noisy channels. *Inf. Comput.*, 206(2-4):378–401, February 2008. ISSN 0890-5401. 1.2

[8] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, February 1981. ISSN 0001-0782. doi: 10.1145/358549.358563. 1.1.1

[9] P.-N. Chen. General formulas for the neyman-pearson type-ii error exponent subject to fixed and exponential type-i error bounds. *IEEE Trans. Inf. Theor.*, 42(1):316–323, September 2006. ISSN 0018-9448. doi: 10.1109/18.481811. 2.3.2

[10] S. Chen, R. Wang, X. Wang, and K. Zhang. Side-channel leaks in web applications: A reality today, a challenge tomorrow. In *Proceedings of the 2010 IEEE Symposium*

on Security and Privacy, SP '10, pages 191–206, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4035-1. doi: 10.1109/SP.2010.20. 1, 1.1.1

[11] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA, 1991. ISBN 0-471-06259-6. 1.2, 5, 2.4.2, 3.1.3, 3.3

[12] J. Demme, R. Martin, A. Waksman, and S. Sethumadhavan. Side-channel vulnerability factor: A metric for measuring information leakage. In *Proceedings of the 39th Annual International Symposium on Computer Architecture*, ISCA '12, pages 106–117, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-1-4503-1642-2. 1.2

[13] J.-F. Dhem, F. Koeune, P.-A. Leroux, P. Mestré, J.-J. Quisquater, and J.-L. Willems. A practical implementation of the timing attack. In *Proceedings of The International Conference on Smart Card Research and Applications*, CARDIS '98, pages 167–182, London, UK, UK, 2000. Springer-Verlag. ISBN 3-540-67923-5. 1, 1.1.2, 4

[14] C. Díaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In *Proceedings of the 2Nd International Conference on Privacy Enhancing Technologies*, PET'02, pages 54–68, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-00565-X. 1.2

[15] W. Diffie and M.E. Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, Nov 1976. ISSN 0018-9448. doi: 10.1109/TIT.1976.1055638. 1, 4

[16] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, SSYM'04, pages 21–21, Berkeley, CA, USA, 2004. USENIX Association. 1.1.1, 1.3.1

[17] K. Dogan, E. Jan, and B. Roland. Stop-and-go-mixes providing probabilistic anonymity in an open system. In *IN PROCEEDINGS OF INFORMATION HIDING WORKSHOP (IH*, pages 83–98. Springer-Verlag, 1998. 1.2

[18] N. S. Evans, R. Dingledine, and C. Grothoff. A practical congestion attack on tor using long paths. In *Proceedings of the 18th Conference on USENIX Security Symposium*, SSYM'09, pages 33–50, Berkeley, CA, USA, 2009. USENIX Association. 1.2

[19] E. W. Felten and M. A. Schneider. Timing attacks on web privacy. In *Proceedings of the 7th ACM Conference on Computer and Communications Security*, CCS '00, pages 25–32, New York, NY, USA, 2000. ACM. ISBN 1-58113-203-4. doi: 10.1145/352600.352606. 1.1.1

[20] J. Friedman. Tempest: A signal problem. *NSA Cryptologic Spectrum*, 1972. 1

[21] D. Genkin, A. Shamir, and E Tromer. RSA key extraction via low-bandwidth acoustic cryptanalysis. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 444–461, 2014. doi: 10.1007/978-3-662-44371-2-25. 1.1.2

[22] J. Ghaderi and R. Srikant. Towards a theory of anonymous networking. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, March 2010. doi: 10.1109/INFCOM.2010.5462155. 1.2

[23] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel. Mutual information analysis. In *Proceeding Sof the 10th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '08, pages 426–442, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-85052-6. doi: 10.1007/978-3-540-85053-3_27. 1.2

[24] E. Gilbert and K. Karahalios. Predicting tie strength with social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 211–220, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518736. 2

[25] V. D. Gligor. *U.S.A. National Computer Security Center, A Guide to Understanding Covert Channel Analysis of Trusted Systems,no. NCSCTG-030 in NSA/NCSC Rainbow Series (Light Pink Book)*. NSA/NCSC Rainbow Series (Light Pink Book). National Security Agency/National Computer Security Center, Fort George G. Meade, MD, USA, 1993. 1.3.2

[26] X. Gong, N. Kiyavash, and N. Borisov. Fingerprinting websites using remote traffic analysis. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 684–686, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0245-6. doi: 10.1145/1866307.1866397. 1.3.1

[27] X. Gong, N. Kiyavash, and P. Venkitasubramaniam. Information theoretic analysis of side channel information leakage in fcfs schedulers. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pages 1255–1259, July 2011. doi: 10.1109/ISIT.2011.6033737. 1.2, 1.3.1, 1.4, 3, 3.1.3, 3.1.4, 7, 3.2.2

[28] X. Gong, N. Borisov, N. Kiyavash, and N. Schear. Website detection using remote traffic analysis. In *Proceedings of the 12th International Conference on Privacy Enhancing Technologies*, PETS'12, pages 58–78, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-31679-1. doi: 10.1007/978-3-642-31680-7-4. 1, 1.1.1, 1.2, 3

[29] G. Greenwald. *No Place to Hide: Edward Snowden, the NSA, and the U.S. Surveillance State*. Henry Holt and Company, 2014. ISBN 9781627790741. 2

[30] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris. *Fundamentals of Queueing Theory*. Wiley-Interscience, New York, NY, USA, 4th edition, 2008. ISBN 047179127X, 9780471791270. 3.2.1

[31] G.H. Hardy. *Ramanujan: Twelve Lectures on Subjects Suggested by His Life and Work*. AMS Chelsea Publishing Series. AMS Chelsea Pub., 1999. ISBN 9780821820230. 4.3.1

[32] I. A. Ibragimov. A note on the central limit theorem for dependent random variables. *Theory of Probability and its Applications*, 20:135–141, 1975. 4.2.2

[33] C.-H. Jong. *Private Communication Detection via Side-channel Attacks*. PhD thesis, University of Maryland at College Park, 2012. 2, 2.2

[34] C.-H. Jong and V. D. Gligor. Private communication detection: A stochastic approach. In *Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WISEC '12, pages 75–86, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1265-3. doi: 10.1145/2185448.2185459. 1, 1.1.1, 2, 2.2

[35] C. H. Jong and V. D. Gligor. Discovering records of private voip calls without wiretapping. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '12, pages 67–68, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1648-4. doi: 10.1145/2414456.2414495. 1, 1.1.1, 2, 2.2, 2.5.1

[36] S. Kadloor, X. Gong, N. Kiyavash, and P. Venkitasubramaniam. Designing router scheduling policies: A privacy perspective. *Signal Processing, IEEE Transactions on*, 60(4): 2001–2012, April 2012. ISSN 1053-587X. doi: 10.1109/TSP.2011.2182348. 1.2, 3

[37] P. C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '96, pages 104–113, London, UK, UK, 1996. Springer-Verlag. ISBN 3-540-61512-1. 1, 1.1.2, 1.1.2, 4, 4.5.1

[38] B. Köpf and D. Basin. An information-theoretic model for adaptive side-channel attacks. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, pages 286–296, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-703-2. doi: 10.1145/1315245.1315282. 1.2

[39] B. Köpf and M. Durmuth. A provably secure and efficient countermeasure against timing attacks. In *Computer Security Foundations Symposium, 2009. CSF '09. 22nd IEEE*, pages 324–335, July 2009. doi: 10.1109/CSF.2009.21. 1.1.2, 1.2

[40] B. Köpf and G. Smith. Vulnerability bounds and leakage resilience of blinded cryptography under timing attacks. In *Proceedings of the 2010 23rd IEEE Computer Security Foundations Symposium*, CSF '10, pages 44–56, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4082-5. doi: 10.1109/CSF.2010.11. 1.2

[41] G. Kramer. *Directed Information in Channels with Feedback*. PhD thesis, Swiss Federal Institute of Technology Zurich, 1970. 3.3, 3.3.1, 3.3.4

[42] J. L. Massey. Causality, feedback, and directed information. In *IEEE International Symposium on Information Theory and Its Applications (ISITA)*, 1990. 1.3.2

[43] S. Mathur and W. Trappe. Bit-traps: Building information-theoretic traffic privacy into packet streams. *Information Forensics and Security, IEEE Transactions on*, 6(3):752–762, Sept 2011. ISSN 1556-6013. doi: 10.1109/TIFS.2011.2138696. 1.2

[44] A. J. Menezes, S. A. Vanstone, and P. C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1996. ISBN 0849385237. 1.1.2

[45] H. Mizuno, K. Iwai, H. Tanaka, and T. Kurokawa. Information theoretical analysis of side-channel attack. In Aditya Bagchi and Indrakshi Ray, editors, *Information Systems Security*, volume 8303 of *Lecture Notes in Computer Science*, pages 255–269. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-45203-1. doi: 10.1007/978-3-642-45204-8-20. 1.2

[46] P. L. Montgomery. Modular Multiplication without Trial Division. *Mathematics of Computation*, 44(170):519–521, 1985. 1.1.2, 4

[47] S. J. Murdoch and G. Danezis. Low-cost traffic analysis of tor. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, SP '05, pages 183–195, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2339-0. doi: 10.1109/SP.2005.12. 1, 1.1.1, 1.2

[48] M. Muresan. *A Concrete Approach to Classical Analysis*. CMS Books in Mathematics. Springer New York, 2015. ISBN 9780387789330. 3.2.1, 4.4.2

[49] M. Naghshvar and T. Javidi. Sequentiality and adaptivity gains in active hypothesis testing. *Selected Topics in Signal Processing, IEEE Journal of*, 7(5):768–782, Oct 2013. ISSN 1932-4553. doi: 10.1109/JSTSP.2013.2261279. 4.3.1

[50] M. Perényi and S. Molnár. Enhanced skype traffic identification. In *Proceedings of the 2Nd International Conference on Performance Evaluation Methodologies and Tools*, ValueTools '07, pages 26:1–26:9, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). ISBN 978-963-9799-00-4. 1.1.1

[51] A. Pfitzmann and M. Hansen. Anonymity, unlinkability, unobservability, pseudonymity, and identity management – a consolidated proposal for terminology. Technical report, 2005. 1.1.1

[52] M.G. Reed, P.F. Syverson, and D.M. Goldschlag. Anonymous connections and onion routing. *Selected Areas in Communications, IEEE Journal on*, 16(4):482–494, May 1998. ISSN 0733-8716. doi: 10.1109/49.668972. 1.1.1

[53] M. K. Reiter and A. D. Rubin. Anonymous web transactions with crowds. *Commun. ACM*, 42(2):32–48, February 1999. ISSN 0001-0782. doi: 10.1145/293411.293778. 1.1.1

[54] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978. ISSN 0001-0782. doi: 10.1145/359340.359342. 1, 4

[55] S. T. Saponas, J. Lester, C. Hartung, S. Agarwal, and T. Kohno. Devices that tell on you: Privacy trends in consumer ubiquitous computing. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, SS'07, pages 5:1–5:16, Berkeley, CA, USA, 2007. USENIX Association. ISBN 111-333-5555-77-9. 1.1.1

[56] H. Sato, D. Schepers, and T. Takagi. Exact analysis of montgomery multiplication. In *Proceedings of the 5th International Conference on Cryptology in India*, INDOCRYPT'04, pages 290–304, Berlin, Heidelberg, 2004. Springer-Verlag. ISBN 3-540-24130-2, 978-3-540-24130-0. doi: 10.1007/978-3-540-30556-9-23. 4, 4.2, 1, 4.2.1

[57] W. Schindler. A timing attack against rsa with the chinese remainder theorem. In *Proceedings of the Second International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '00, pages 109–124, London, UK, UK, 2000. Springer-Verlag. ISBN 3-540-41455-X. 4, 4.2, 4.2.1, 10

[58] W. Schindler. On the optimization of side-channel attacks by advanced stochastic methods. In *Proceedings of the 8th International Conference on Theory and Practice in Public Key Cryptography*, PKC'05, pages 85–103, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-24454-9, 978-3-540-24454-7. doi: 10.1007/978-3-540-30580-4-7. 1.1.2, 1.2, 1.4, 4, 4.2, 1, 2

[59] W. Schindler. Exclusive exponent blinding may not suffice to prevent timing attacks on RSA. In *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, pages 229–247, 2015. doi: 10.1007/978-3-662-48324-4-12. 1.2, 4.5.1

[60] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *Proceedings of the 2Nd International Conference on Privacy Enhancing Technologies*, PET'02, pages 41–53, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-00565-X. 1.2

[61] S. Shintre, V. D. Gligor, and J. Barros. Anonymity leakage in private voip networks. *to appear in IEEE Trans. Dependable and Secure Computing*, 2015. 1.2

[62] D. X. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on ssh. In *Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10*, SSYM'01, Berkeley, CA, USA, 2001. USENIX Association. 1.1.1

[63] C. D. Walter. Precise bounds for montgomery modular multiplication and some potentially insecure rsa moduli. In *Proceedings of the The Cryptographer's Track at the RSA Conference on Topics in Cryptology*, CT-RSA '02, pages 30–39, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-43224-8. 1

[64] C. D. Walter and S. Thompson. Distinguishing exponent digits by observing modular subtractions. In *Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA*, CT-RSA 2001, pages 192–207, London, UK, UK, 2001. Springer-Verlag. ISBN 3-540-41898-9. 4.2, 1

[65] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz. Primary user behavior in cellular networks and implications for dynamic spectrum access, 2012. 2.1.2

[66] P. Wright and P. Greengrass. *Spycatcher: The Candid Autobiography of a Senior Intelligence Officer*. A Dell Book. Dell, 1987. ISBN 9780440201328. 1

[67] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart. Cross-vm side channels and their use to extract private keys. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 305–316, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1651-4. doi: 10.1145/2382196.2382230. 1

[68] Y. Zhu, Y. Lu, A. Vikram, and H. Fu. On privacy of skype voip calls. In *Proceedings of the 28th IEEE Conference on Global Telecommunications*, GLOBECOM'09, pages 5735–5740, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-4147-1. 1.1.1