RSA Laboratories

Volume 6, No. 1, Spring 2003

CryptoBytes

CONTENTS

I. Radio-Frequency Identification: Security Risks and Challenges

Sanjay E. Sarma, Stephen A. Weis and Daniel W. Engels

ABSTRACT

Low-cost Radio Frequency Identification (RFID) tags affixed to consumer items as "smart-labels" may emerge as one of the most pervasive computing technologies in history. While RFID systems can yield great productivity gains, they may also expose new threats to the security and privacy of both individuals and organizations. This article gives a brief introduction to RFID technology and describes potential security and privacy risks. We present several challenges to providing desired security properties in the unique setting of low-cost RFID devices and identify several areas for future research.

II. Identity-Based Encryption: a Survey

Martin Gagné

ABSTRACT

Identity-Based Encryption is a form of public key encryption for which the public key can be an arbitrary string, and in particular, a string that identifies the user who holds the associated private key, like his email address. The original motivation for identity-based cryptography was to simplify certificate management, but it has many other applications. In this paper, we survey recent proposals for usable identity-based encryption schemes.

III. Advances in Side-Channel Cryptanalysis, Electromagnetic Analysis and Template Attacks

Dakshi Agrawal, Bruce Archambeault, Suresh Chari, Josyula R. Rao and Pankaj Rohatgi

ABSTRACT

We describe two recent advances which substantially increase the scope and power of side-channel cryptanalysis. The first advance is the exploitation of information leakage from electromagnetic emanations. The second advance, known as template attacks, is a superior data analysis technique which substantially reduces the number of side-channel samples needed for an attack. These advances pose a risk to all cryptographic implementations, including those immune against earlier side-channel attacks.

RSA LABORATORIES

Radio-Frequency Identification: Security Risks and Challenges*

Sanjay E. Sarma, Stephen A. Weis and Daniel W. Engels

Abstract

Low-cost Radio Frequency Identification (RFID) tags affixed to consumer items as "smart-labels" may emerge as one of the most pervasive computing technologies in history. While RFID systems can yield great productivity gains, they may also expose new threats to the security and privacy of both individuals and organizations. This article gives a brief introduction to RFID technology and describes potential security and privacy risks. We present several challenges to providing desired security properties in the unique setting of low-cost RFID devices and identify several areas for future research.

1 Introduction

Automatic Identification (Auto-ID) systems are a common tool in manufacturing processes, just-in-time inventory control, logistics and point of sale product identification. For over twenty years, the bar code has been a familiar optical Auto-ID system found on many consumer items. Perhaps the most common bar code is the linear, or one-dimensional, Universal Product Code (UPC) designed in 1973 [24]. The US Postal Service and several commercial shipping companies have also adopted two-dimensional bar codes, which are able to carry more data. More recently, Radio Frequency Identification (RFID) systems have made inroads into retail logistics markets. In the near future, low-cost RFID "smart-labels" may become an economical, and efficient replacement for optical bar codes.

Radio Frequency Identification systems have emerged as a practical Auto-ID platform in industries as varied as automobile manufacturing, microchip fabrication, even cattle herding. RFID systems are composed of radio frequency (RF) tags, or transponders, and RF tag readers, or transceivers. Tag readers broadcast an RF signal to access resident data stored on tags, typically including a unique identification number.

Most tags consist of an antenna or other coupling element connected to an integrated circuit, allowing the incorporation of tag functionality such as writable storage, environmental sensors, access control or encryption. Simple RFID devices may be found everyday in keyless entry systems, automated tollbooths, subway stations or in clothing. Several examples of RFID tags appear in Figure 1.

RFID tags offer several advantages over optical bar codes. Data may be read automatically: without line of sight, through non-conducting material such as cardboard or paper, at a rate of several hundred tags per second and from a distance of several meters. Given that optical barcodes are scanned over 5 billion times daily [24], efficiency gains from using RFID tags could substantially lower the cost of tagged items.

*Work carried out at the MIT Auto-ID Center and Laboratory for Computer Science. Authors may be reached at: {sesarma, sweis, dwe}@mit.edu

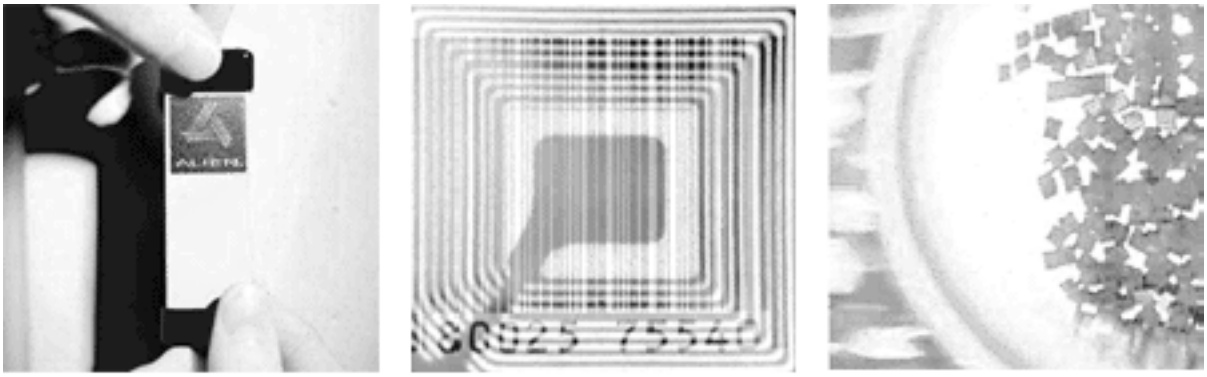

Figure 1: An RFID tag, an RFID tag with printed barcode and dust-sized RFID microchips.

By integrating a unified identification system on all levels in the supply chain, every party involved in the lifespan of a product may reap the benefits of an RFID based object identification system. This includes not only manufacturers and retailers, but also consumers, regulatory bodies such as the United States Food and Drug Administration, and even waste disposal firms. The potential cost savings are likely to make RFID tags one of the most widely deployed microchips in history.

Unfortunately, the universal deployment of RFID tags in consumer items may create new security issues not present in closed manufacturing environments. Consumer privacy may be compromised by nearby attackers extracting data from unprotected tags. Individuals may be physically detected by associating their identities with tags they carry, violating "location privacy" similar to an issue in Bluetooth [11]. Corporate espionage is another risk: A retailer's inventory labeled with unprotected RFID tags could be monitored or tracked by competitors; yielding valuable sales and marketing data.

Presently, most deployed RFID systems are used for identification of higher value items, such as microchips or automobile components. In these industries, tags costing US$0.50-US$1.00 or more are economical. At these costs, tags may be equipped with resources to support strong cryptographic primitives, tamper resistant packaging or other security enhancing features. Many developments relevant to smart card research are applicable to these higher value RFID tags.

However, significant consumer market penetration will occur only if tags are priced below US$0.10 and can be incorporated into most paper packaging. Under this price ceiling, supporting strong cryptographic primitives is presently not a viable option. Even if silicon manufacturing developments allow more features to be included in a low-cost, US$0.05 tag, there will be continual pressure from high volume customers for even lower cost tags.

This is due to the economics of the RFID tag market. Tag purchases will likely be made by manufacturers in high volumes. Even slight differentials in tag design costs translate into large financial savings. Consider the recent purchase of 500 million low-cost tags [18] by a major consumer product manufacturer. A difference of US$0.01 between two tag design costs represents US$5,000,000 in savings. Tags supporting greater functionality, such as strong cryptographic primitives, must offer tag owners enough economic benefits to justify their higher costs.

The design of low-cost RFID systems is part of ongoing research at the MIT Auto-ID Center [3]. An overview of RFID systems and their security issues is available in [21]. Proposals addressing several of these issues are presented in [26]. As mentioned, the resource-starved environment of low-cost RFID tags is most closely related to issues explored in the context of smart cards. Particularly relevant are cost and security trade-offs of smart cards, presented in [1].

RFID tags will frequently operate in insecure environments, possibly subjected to intense physical attacks. Discussion of smart card operation in hostile environments is presented in [9], and a comprehensive overview of many physical attacks and countermeasures appears in [25]. Specific low-cost attacks appear in [2] and are part of ongoing research at the University of Cambridge's TAMPER Lab [23]. Cautionary information regarding the implementation of AES in smart cards is available in [7]. Due to passive powering and a wireless interface, RFID tags may be especially susceptible to fault induction, timing attacks or power analysis attacks, highlighted in [4, 15, 14], and [13].

In this article, Section 2 provides a brief introduction to the components and operation of RFID systems. Section 3 states assumptions about design features, performance requirements and resource limitations. Section 4 focuses on the security and privacy aspects of RFID systems, detailing risks and challenges in Sections 4.1 and 4.2, respectively.

2 RFID Primer

RFID systems are composed of three key elements:

- the RFID tag, or *transponder*, carries object identifying data.
- the RFID reader, or *transceiver*, interfaces with tags to read or write tag data.
- the back-end database aggregates and utilizes tag data collected by readers.

All items to be identified in an RFID system are physically labeled with a tag. Tags are typically composed of a microchip for data storage and logical operations, and a coupling element, such as an antenna coil, used to communicate to readers via radio frequencies (RF). In addition, tags may contain a direct contact interface as found in smart cards. Tag memory may be a read-only, write-once read-many or be fully rewritable.

Tag readers *interrogate* tags for their data through an RF interface. To provide additional functionality, readers may also contain internal storage, processing power or connections to back-end databases. Computation may be carried out by readers on behalf of tags, particularly in cryptographic applications.

Tags may either be *actively* powered through an on-board power source such as a battery, or *passively* powered. Passive tags inductively receive power through an RF signal from the reader. To offer an analogy for this process, one may think of readers as "shouting" out to passive tags, then extracting data from the resultant echoes. Extending this analogy, the reader's shouts may be monitored by eavesdroppers from a greater range than the tag's echoes. We will explore this issue further in Section 4.

The maximum distance that a reader can communicate with a tag is determined by the type of tag. Active tags can boost reply signals with on-board power and reply to readers at a greater range than passive tags. Active tags may carry an on-board clock and perform calculations or take sensor readings in the absence of a reader. Passive tags may only operate in the presence of a reader and are inactive otherwise.

Readers may use tag contents as a look-up key into a back-end database. The back-end database may associate product information, tracking logs, or key

management data with particular tags. Independent databases may be built by anyone with access to tag contents, allowing unrelated users along the supply chain to develop their own applications.

To illustrate the interaction of RFID system components, consider an example application of an RFID-enabled warehouse. Each item in the warehouse would be labeled with an RFID tag containing identifying information such as the manufacturer, product type and a unique serial number. These contents may be referred to collectively as the tag ID and in practice could be represented by 96 bits.

Shelves, forklifts and doorways in the warehouse would each be equipped with an RFID tag reader. Shelves would "know" which items they contained and when items were removed. Similarly, forklifts would know which items they were carrying and doors would know which items passed through.

Each of these transactions would be recorded by the tag readers in a back-end database, creating an account of the entire history and whereabouts of a particular item. External information, such as shipping or purchasing data may be associated with a particular item's record in the back-end database.

Many applications for this type of data are apparent. For example, a RFID-enabled warehouse could take an instant inventory of its contents. Items could be located instantly, yet could be stored or moved at any time. Finally, preventing "shrinkage" (an industry euphemism for theft) and identifying its culprits could be aided by this RFID-enabled warehouse.

3 Design Assumptions

The narrow cost requirements of low-cost RFID systems make low-cost tags extremely resource scarce environments. A practical 2-3 year estimate of the security resources available to a US$0.05 design, such as those proposed by the MIT Auto-ID Center [3, 20], may be limited to hundreds of bits of storage, roughly 5,000-10,000 gates and a max communication range of a few meters. Such low-cost tags will almost certainly be passively powered, due to costs associated with active power sources.

A US$0.05 tag in 2003 may have approximately 250-1000 gates available for security features. Furthermore, security protocols and computations must allow for read rates of hundreds per second. Depending on the particular tag implementation, power consumption may be another limiting factor.

These limits are far below the requirements for a public-key cryptographic system, even a resource efficient scheme such as NTRU [10, 16]. Most symmetric encryption algorithms are also beyond available tag resources. For example, commercial AES implementations typically have on the order of 20,000-30,000 gates [6], which is more than is available for the entire low-cost tag design. Even the hardware implementations of standard cryptographic hash functions such as SHA-1 are currently too costly [6].

We assume tags have insecure memories whose entire contents may be extracted by physical attacks as described in [25]. These attacks may include laser or water etching, X-ray or ion probing, TEMPEST attacks, clock glitching, circuit disruption or many other varied attacks. Luckily, these attacks require physical tag access and are likely to be detected when carried out in public. Privacy is rather a moot point if someone can surreptitiously obtain a tagged item, conduct a physical attack, then return the item without detection. The important implication is that RFID tags cannot be trusted to securely store long-term secrets, such as shared keys, when left in isolation.

The reader-to-tag, or *forward* channel, may be monitored from a great distance. Depending on the implementation, the relatively weaker tag-to-reader, or

backward channel, may also be monitored from a considerable distance. This largely depends on the communication frequency between tags and readers. At 900 MHz, eavesdroppers could theoretically monitor the forward channel from 1 kilometer and the backward channel from up to 100 meters. Fortunately, in practice attaining these ranges would be difficult. The 13.56 MHz and 2.45 GHz operating frequencies also have asymmetric eavesdropping ranges, but from shorter distances. It should be noted that these ranges are for non-interactive eavesdropping only and that any active communication with tags must occur within a short distance, perhaps 2 meters.

Tags may be equipped with a physical contact channel for critical functions or "imprinting" tags with secret keys [22]. Additionally, we may assume the tag packaging contains a barcode, human-readable digits or other information to corroborate tag data, as in the design presented in [12]. To further authenticate tags, readers may measure physical properties such as tag signal power levels or response times, as used in [17]. This may act as a countermeasure against spoofing attempts.

It is assumed that a secure connection exists between tag readers and the back-end database. Tag readers may also perform cryptographic calculations or interface with key management systems on behalf of tags. Tags may be assumed to have a mechanism to reveal their presence, called a *ping*. Anyone may ping a tag, which will respond with some non-identifying signal. Finally, tags will be equipped with a *kill* command rendering them permanently inoperable. The kill command may be designed to be a slow operation which physically disables the tag, perhaps by disconnecting the antenna or shorting a fuse.

4 Security and Privacy

4.1 Risks

Privacy is a major issue in a ubiquitous RFID system. Consumer products labeled with insecure tags may reveal sensitive information when queried by nearby snoops. Most consumers prefer to keep their brand of RFID-tagged underwear or prescription medicine private from nosy passersby. Retail businesses also may be threatened by unauthorized readers. For example, a corporate spy could periodically take inventory of a store's shelves to infer sales data.

A related threat is that of tracking, or violations of "location privacy". Concerns over location privacy were recently raised when a major tire manufacturer announced plans to embed RFID tags in their products [19]. Although the tag contents may be secured, predictable tag responses could allow tags to be associated with their holders' identity. Even if tags do not leak unique identifying information individually, a set of tags may be tracked as a "constellation"; a distinct taste in brands could betray someone's identity. Individuals carrying tags may be tracked as they pass by fixed tag readers. Corporate spies might be able to derive valuable logistics information from insecure RFID tagged packages, even without knowing the actual contents of the package themselves.

Denial of service is another threat. Attackers could attempt to jam RF signal channels, or to disable tags by some other means. This is especially relevant to the retail market, where there is an interest in RFID-enabled automatic checkout. Theft may result if tags are able to be "cloaked" from store readers. Of course, preventing low-tech attacks, such as dropping an item into a metal lined bag, would require traditional countermeasures such as security guards or cameras.

Thieves should not be able to effectively spoof tags, either. A thief could create their own tag to spoof a valid item. By replacing actual items with these decoy tags, a thief might fool a shelf that the valid items were still in stock. Alternatively, a thief could attempt to rewrite valid RFID tag contents so that they represented lower value items at checkout.

It should be noted that these risks are most important on a widespread scale. Existing bar code systems are publicly readable, can be spoofed or disabled, and therefore exhibit the same risks. However, these attacks do not have the potential to be carried out wirelessly, on a massive scale. The challenges of addressing these issues is to prevent wide-scale or automated attacks made feasible by RFID's efficient wireless interface, without exceeding narrow cost barriers.

4.2 Challenges

The primary challenge in providing privacy and access control mechanisms in low-cost RFID is scarcity of resources. As mentioned, tags will only have a fraction of the gate count available in smart cards. Security mechanisms of passively powered tags will need to be carefully designed so as not to leave tags in an insecure state in the event of power loss or interruption.

Additionally, security protocols must account for the asymmetric signal strength between the forward and backward channels. For example, anti-collision algorithms carried out by tag readers addressing multiple tags may leak data over the "loud" forward channel, a threat described in [26].

Low-cost tag security mechanisms typically are not expected to be resilient to lengthy, determined attacks. Recall that attacks would need to originate from within the short (e.g., 2-meter) operating range of a tag, making it easier to detect in a retail setting. Anecdotally, low-cost tags used by retailers might be required to be resistant to protocol attacks (i.e., not relying on physical or electromagnetic properties) for approximately 10 minutes. Making a liberal assumption that tags can support 1000 commands per second, this represents 600,000 brute force attempts.

Under these constraints, a simple access control "lock" mechanism based on hash functions has been proposed in [21] and [26]. A challenge to the research community is to provide hardware-efficient cryptographic hash functions within low-cost RFID tag resource constraints. Low-cost symmetric encryption schemes are another desirable primitive. The aptly named "Tiny Encryption Algorithm" [27, 28], which has a small implementation size relative to DES or AES, may be a step in the right direction.

To address location privacy, tags cannot respond to queries in a predictable manner. This motivates inclusion of an on-board random number generator to randomize tag responses, as in the extension of the hash-lock design from [26]. The further development of practical low-cost pseudo-random number generators, or sources of physical randomness [8] would benefit RFID designs, as well as many other applications. Hardware-efficient perfect one-way hash functions [5] would be another useful primitive to counter tag tracking.

Regardless of the underlying mechanisms providing privacy and access control, management of tag keys is an important issue. Initialization, storage and transfer of keys should be economical. Since tags may pass through the hands of manufacturers, retailers and consumers, there should exist an efficient means to transfer tag ownership. In some settings, physical possession of a tag may confer tag ownership, perhaps through a contact channel or some optical information on a tag. In other scenarios, such as a rental setting, some external key data must represent tag ownership. Providing flexible access control and key management tools at a reasonable cost is a challenge to the design of tags, readers and back-end databases.

Flexibility and openness of design are of utmost importance to a successful RFID system. Future tag developments will allow greater storage, faster performance and new functionality to be incorporated into low-cost tag designs. Current security mechanisms for RFID systems should not impede utilization of future technologies, nor should they adversely affect the user experience. Retailers will not adopt RFID systems if they necessarily hinder the consumer check-out process. No one expects customers to undergo complicated security procedures every time they purchase a quart of milk.

On the other hand, RFID tags should be as open a platform as possible, supporting both existing applications and applications yet to be conceived. Security features should not interfere with the development of new applications by third parties. Many useful consumer applications could emerge from grass-roots development and should not be impaired by proprietary or closed security mechanisms.

5 Conclusion

The success of a consumer RFID system may depend on developing appropriate tools for providing security and privacy. Hardware efficient hash functions, symmetric encryption and random number generators all play a crucial role in developing the RFID security mechanisms necessary to dissuade large-scale attacks. New protocols resilient to eavesdropping, fault induction, or power analysis while maintaining performance and costs will be a valuable area of research. Integrating RFID systems with a key management infrastructure is another issue requiring further development.

In many ways, a universal low-cost RFID system is a precursor to ubiquitous computing. Improving technology and allowing integration of more features will blur the line between RFID tags, smart cards and general purpose computers. Security lessons learned from RFID systems will benefit the development of secure ubiquitous computing systems of the future.

6 Acknowledgments

Images courtesy of the MIT Auto-ID Center, Checkpoint Systems and Alien Technology. Thanks to Ron Rivest and Peter Cole for their support and input. Thanks to Lawrence Gold for review and comments.

References

[1] Martin Abadi, Michael Burrows, C. Kaufman, and Butler W. Lampson. Authentication and Delegation with Smart-cards. In *Theoretical Aspects of Computer Software*, pages 326–345, 1991.

[2] Ross Anderson and Markus Kuhn. Low Cost Attacks on Tamper Resistant Devices. In *IWSP*. LNCS, 1997.

[3] Auto-ID Center. http://www.autoidcenter.org.

[4] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. In *EUROCRYPT'97*, volume 1233, pages 37–51. LNCS, Advances in Cryptology, 1997.

[5] Ran Canetti, Daniele Micciancio, and Omer Reingold. Perfectly One-Way Probabilistic Hash Functions. In *ACM Symposium on Theory of Computing*, pages 131–140, 1998.

[6] CAST Inc. AES and SHA-1 Cryptoprocessor Cores. http://www.cast-inc.com.

[7] Suresh Chari, Charanjit Jutla, Josyula R. Rao, and Pankaj Rohatgi. A Cautionary Note Regarding Evaluation of AES Candidates on Smart-Cards. In *Second Advanced Encryption Standard (AES) Candidate Conference*, Rome, Italy, 1999.

[8] Blaise Gassend, Dwaine Clarke, Marten van Dijk, and Srinivas Devadas. Controlled Physical Random Functions. In *Computer Security Conference*, December 2002.

[9] Howard Gobioff, Sean Smith, J. Doug Tygar, and Bennet Yee. Smart Cards in Hostile Environments. In *Workshop on Elec. Commerce*, 1996.

[10] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. *LNCS*, 1423:267–283, 1998.

[11] Markus Jakobsson and Susanne Wetzel. Security Weaknesses in Bluetooth. *LNCS*, 2020:176+, 2001.

[12] Ari Juels and Ravikanth Pappu. Squealing Euros: Privacy Protection in RFID-Enabled Banknotes. In *Financial Cryptography*, 2003.

[13] Burton S. Kaliski Jr and Matt J. B. Robshaw. Comments on Some New Attacks on Cryptographic Devices. RSA Laboratories' Bulletin No. 5, July 1997. http://www.rsasecurity.com/.

[14] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. *LNCS*, 1666:388–397, 1999.

[15] Paul C. Kocher. Cryptanalysis of Diffie-Hellman, RSA, DSS, and other Systems Using Timing Attacks. Technical report, Cryptography Research, Inc., 1995.

[16] NTRU. GenuID. http://www.ntru.com/.

[17] Adrian Perrig, Ran Canetti, J.D. Tygar, and Dawn Song. The TESLA Broadcast Authentication Protocol. *RSA CryptoBytes*, 5(2):2–13, Summer/Fall 2002.

[18] RFID Journal. Gillette to Purchase 500 Million EPC Tags. http://www.rfidjournal.com, November 2002.

[19] RFID Journal. Michelin Embeds RFID Tags in Tires. http://www.rfidjournal.com, January 2003.

[20] Sanjay E. Sarma. Towards the Five-Cent Tag. Technical Report MIT-AUTOID-WH-006, MIT Auto-ID Center, 2001.

[21] Sanjay E. Sarma, Stephen A. Weis, and Daniel W. Engels. RFID Systems and Security and Privacy Implications. In *CHES*, pages 454–470. LNCS, 2002.

[22] Frank Stajano and Ross Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *IWSP*, volume 1796, pages 172–194. LNCS, 1999.

[23] TAMPER Lab. University of Cambridge Tamper and Monitoring Protection Engineering Research Lab. http://www.cl.cam.ac.uk/Research/.

[24] Uniform Code Council. Homepage. http://www.uc-council.org/.

[25] Steve H. Weingart. Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defences. In *CHES*, volume 1965, pages 302–317. LNCS, 2000.

[26] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In *Security in Pervasive Computing*, 2003.

[27] David J. Wheeler and Robert M. Needham. TEA, a Tiny Encryption Algorithm. Technical report, Computer Laboratory, University of Cambridge, 1995.

[28] David J. Wheeler and Robert M. Needham. TEA Extensions. Technical report, Computer Laboratory, University of Cambridge, 1997.

Identity-Based Encryption: a Survey

Martin Gagné *
gagne@cs.ucdavis.edu

Abstract

Identity-based encryption is a form of public key encryption for which the public key can be an arbitrary string, and in particular, a string that identifies the user who holds the associated private key, like his email address. The original motivation for identity-based cryptography was to simplify certificate management, but it has many other applications. In this paper, we survey recent proposals for usable identity-based encryption schemes.

1 Introduction

The concept of Identity-Based Encryption (IBE) was first formulated by Shamir in 1984 [28]. In such a scheme, the public key can be an arbitrary string. For example, if Alice wants to send a message to Bob at bob@yahoo.com, then she simply encrypts the message using the string "bob@yahoo.com" as the public key. The original motivation for this idea was to eliminate the need for directories and certificates by using the identity of the receiver as the public key, but it can also be used to implement ephemeral (short lived) public keys, manage user credentials, or for the delegation of decryption keys. Recently, it has also been used to build forward-secure encryption schemes. Efficient solutions for the related notion of identity-based signatures were quickly found ([13, 12]), but identity-based encryption proved to be much more challenging. Most schemes proposed since 1984 ([9, 31, 30, 24, 20]) were unsatisfactory because they were too computationally intensive, they required tamper resistant hardware, or they were not secure if users colluded.[1] In this paper, we survey recent proposals of identity-based encryption schemes which do not suffer from any of these drawbacks [7, 2], and some variants of [2] which provide additional functionality [22, 17].

Informally, an identity-based encryption scheme consists of four algorithms: (1) **Setup** generates the system parameters and a master-key, (2) **Extract** uses the master-key to generate the private key corresponding to an arbitrary string $\mathsf{ID} \in \{0,1\}^*$, (3) **Encrypt** encodes a plaintext using a public key ID and (4) **Decrypt** decodes ciphertexts using the corresponding private key. The algorithm **Setup** is run by a trusted authority which we call the private key generator (PKG). The PKG also runs the algorithm **Extract** at the request of a user who wishes to obtain the private key corresponding some string (see Figure 1). Note that the user likely needs to prove to the PKG that he is the legitimate "owner" of this string (for example, to obtain the private key corresponding to "bob@yahoo.com", the user must prove that bob@yahoo.com is truly his email address). The algorithms **Encrypt** and **Decrypt** are run by the users to encrypt and decrypt messages.

*Partially supported by NSF ITR CCR-0205733 and the Packard Foundation.

[1] We are talking here about schemes which do not require an online authority for decryption. Mediated identity-based encryption schemes are relatively easy to build, [1] is a good example.

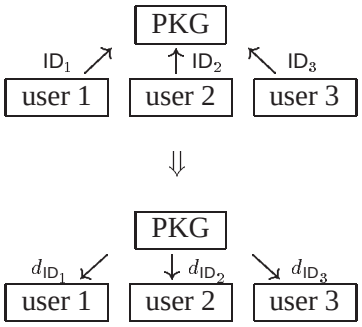

Figure 1: Private key request in an IBE scheme.

The rest of this paper is structured as follows. In Section 1.1, we discuss some applications of identity-based encryption schemes. In Section 2, we present a simple identity-based encryption scheme to introduce the concept. We formally define the notion of identity-based encryption and give the security model in Section 3. In Section 4, we survey the identity-based encryption schemes whose security can be proven in the model of Section 3.

1.1 Applications of Identity-Based Encryption

We already mentioned that the original motivation for identity-based encryption was to simplify certificate management. We now present other applications.

Revocation of public keys. Public key certificates contain a preset expiration date. In an identity-based encryption scheme, we can make the keys expire by encrypting the messages using the public key "receiver-address || current-date" where current-date can be the day, week, month or year depending on the frequency at which we want the users to renew their private key. Note that unlike traditional public key infrastructure, the senders do not need to obtain new certificates every time the private keys are renewed, however, the receiver must query the PKG each time to obtain the new private key. So identity-based encryption is a very efficient way of implementing ephemeral public keys. This is also useful if, for example, the private key is kept on a laptop: if the laptop is stolen, only the private key corresponding to that period of time is compromised, the master-key is unharmed. This approach can also be used to send messages into the future since the receiver will not be able to decrypt the message until he gets the private key for the date specified by the sender from the PKG (see [27, 10] for methods of sending messages into the future using a stronger security model).

Managing user credentials. By encrypting the messages using the address "receiver-address || current-date || clearance-level", the receiver will be able to decrypt the message only if he has the required clearance. This way, the private key generator can be used to grant user credentials. To revoke a credential, the PKG simply stops providing the private key in the next time period.

Delegations of decryption keys. Suppose a manager has several assistants each responsible for a different task. Then the manager can act as the private key generator and give his assistants the private keys corresponding to their responsibilities (so the public key would be 'Duty'). So each assistant can decrypt the messages whose subject fall within its responsibilities, but cannot decrypt messages intended for other assistants. The manager can decrypt all the messages using his master-key.

Forward-secure encryption. Some of the schemes presented in this paper can also be used as building blocks to construct forward-secure encryption schemes ([4]) and key-insulated cryptosystems ([11]). In a forward-secure encryption scheme, the receiver's private key evolves at each time period so that if the private key of a time period is compromised, all the messages encrypted in previous time periods secure[2]. In a key-insulated encryption scheme, the secret key is

[2]However, all the private keys for time periods after the exposure are also compromised.

divided into two parts, both evolving at every time interval, which must be combined to obtain the private decryption key. Future secret keys are compromised only if both parts are exposed in the same time period[3].

2 A Simple Identity-Based Encryption Scheme

We now introduce an identity-based encryption scheme based on quadratic residues proposed by Cocks [7]. This scheme is not as efficient and possibly not as secure as the schemes we present in Section 4, but it is easier to understand and not as involved mathematically. We present it here to introduce the concepts of identity-based encryption.

In the setup phase, the PKG generates two random primes p and q, each congruent to $3 \bmod 4$ such that $N = pq$ is hard to factor. The PKG also picks a cryptographic hash function $H : \{0,1\}^* \to \mathcal{Q}$, where $\mathcal{Q}$ is the set of integers in $\mathbb{Z}_N^*$ whose Jacobi symbol is 1. N and H are then published by the PKG, p and q are kept secret. Note that under these conditions, for any $a \in \mathcal{Q}$, either a or $-a$ is a square modulo N. [4]

To extract the private key corresponding to a string $\mathsf{ID} \in \{0,1\}$, the PKG computes

$$d_{\mathsf{ID}} = H(\mathsf{ID})^{\frac{N+5-(p+q)}{8}} \bmod N$$

and sends d_{ID} to the requesting user. One can easily show that the resulting d_{ID} satisfies either $d_{\mathsf{ID}}^2 = H(\mathsf{ID}) \bmod N$ or $d_{\mathsf{ID}}^2 = -H(\mathsf{ID}) \bmod N$ depending on which of $H(\mathsf{ID})$ or $-H(\mathsf{ID})$ is a square $\bmod N$.

The encryption function encodes the bits of the message one at a time. Given a single bit x encoded as 1 or -1 and the identifying string ID of the receiver, the sender generates random values $t_1, t_2 \in \mathbb{Z}_N^*$, $t_1 \neq t_2$, with Jacobi symbol $\left(\frac{t_i}{N}\right) = x$ and computes $s_1 = t_1 + H(\mathsf{ID})/t_1 \bmod N$ and $s_2 = t_2 - H(\mathsf{ID})/t_2 \bmod N$. The ciphertext is $C = \langle s_1, s_2 \rangle$.

The receiver recovers the bit x from a ciphertext $\langle s_1, s_2 \rangle$ using his private decryption key d_{ID} as follows. First, he picks $s = s_1$ if $d_{\mathsf{ID}}^2 = H(\mathsf{ID})$, otherwise, he picks $s = s_2$. Then, he computes

$$x = \left(\frac{s + 2d_{\mathsf{ID}}}{N}\right).$$

This x is the original bit since $s + 2d_{\mathsf{ID}} \equiv t(1 + d_{\mathsf{ID}}/t)^2 \pmod N$ where t is the t_i value corresponding to s, so

$$\left(\frac{s + 2d_{\mathsf{ID}}}{N}\right) = \left(\frac{t}{N}\right)\left(\frac{1 + d_{\mathsf{ID}}/t}{N}\right)^2 = \left(\frac{t}{N}\right) = x$$

unless $1 + d_{\mathsf{ID}}/t \notin \mathbb{Z}_N^*$ which is extremely unlikely[5].

The security of this scheme is based on the *Quadratic Residuosity Problem* i.e. on the difficulty of determining if whether or not a number $a \in \mathcal{Q}$ is a square modulo N if the factorization of N is unknown. Unfortunately, [7] does not provide a proof of security in a security model as strong as the one we give in Section 3. Also, it is very easy to delete, add or modify bits in the encrypted message, so additional integrity protection must be employed to confirm the validity of the message.

Another drawback of this scheme is the message expansion factor of $2 \log_2 N$ (this would be 2048 in practice nowadays). However, the main use of public key cryptosystems is the exchange of session keys, so, for a 128 bit session key, the total length of the ciphertexts would be 32K bytes, an acceptable overhead for many applications.

[3] See [11] for more details about the security of these schemes.

[4] This is because -1 is a quadratic non-residue both in $\mathbb{Z}_p$ and $\mathbb{Z}_q$.

[5] If $1 + d_{\mathsf{ID}}/t \notin \mathbb{Z}_N^*$, then either $t = -d_{\mathsf{ID}}$, so the sender has guessed the receiver's private key, or $\gcd(1 + d_{\mathsf{ID}}/t, N)$ is a non-trivial factor of N, so the receiver can factor N.

3 Definitions

Identity-Based Encryption. An identity-based encryption scheme consists of four randomized algorithms: **Setup**, **Extract**, **Encrypt** and **Decrypt**.

Setup: takes as input a security parameter and outputs params (system parameters) and master-key. The system parameters must include the description of the message space $\mathcal{M}$ and the ciphertext space $\mathcal{C}$. The system parameters will be publicly known while the master-key is known only to the private key generator (PKG).

Extract: takes as input the system parameters params, the master-key and an arbitrary string $\mathsf{ID} \in \{0,1\}^*$ and outputs the private key d_{ID} corresponding to the public key ID.

Encrypt: takes as input the system parameters params, a public key ID and a plaintext $M \in \mathcal{M}$ and outputs a corresponding ciphertext.

Decrypt: takes as input the system parameters params, a private key d_{ID} and a ciphertext $C \in \mathcal{C}$ and outputs the corresponding plaintext.

These algorithms must satisfy the standard consistency constraints, namely if all the algorithms are applied correctly, then any message in the plaintext space encrypted with the algorithm **Encrypt** should be correctly decrypted by the algorithm **Decrypt**.

We note that each user needs to establish a secure channel with the private key generator when requesting his private key in order to keep the private key secret.

Chosen ciphertext security. A public key encryption scheme is considered secure if an adversary is unable to obtain any information about a ciphertext even if he is given the decryption of any other ciphertext of his choice. The standard definition of security capturing this notion is that of chosen ciphertext security defined by Rackoff and Simon in [26]. However, in our setting the adversary may also be able to obtain the private key corresponding to some IDs of this choice other than the one on which he is being tested. The system should remain secure against such an attack. Therefore, the definition of security must be strengthened a little to allow the adversary to obtain the private key corresponding to any IDs except the one on which he is being tested.

The notion of semantic security against adaptive chosen ciphertext attack for an identity-based encryption scheme is defined through the following game:

- The challenger chooses a security parameter k and runs the **Setup** algorithm. He returns to the adversary the public system parameters params and keeps the master-key to himself.
- The adversary issues queries $q_1, \ldots q_m$ where each query is one of:
  – Extraction query $\langle \mathsf{ID}_i \rangle$. The challenger responds by running the algorithm Extract to generate the private key d_i corresponding to the public key ID_i and sends it to the adversary.
  – Decryption query $\langle \mathsf{ID}_i, C_i \rangle$. The challenger responds by running algorithm Extract to generate the private key d_i corresponding to the public key ID_i, uses the algorithm Decrypt together with this private key to decode the ciphertext C_i and returns the resulting plaintext to the adversary.
- The adversary outputs two equal length plaintexts $M_0, M_1 \in \mathcal{M}$ and a public key ID on which he wishes to be tested. ID must not have appeared in any previous extraction query. The challenger picks a random bit $c \in \{0,1\}$ and sends $C = \textbf{Encrypt}(\mathsf{params}, \mathsf{ID}, M_c)$ as the challenge to the adversary.
- The adversary issues queries $q_{m+1}, \ldots, q_n$ as before, except that he cannot issue the extraction query $\langle \mathsf{ID} \rangle$ or the decryption query $\langle \mathsf{ID}, C \rangle$.

- The adversary outputs a guess $c' \in \{0,1\}$. The adversary wins the game if $c' = c$.

The advantage of an attacker $\mathcal{A}$ against the scheme is defined to be $ADV_{\mathcal{A}}(k) = |Pr[c = c'] - 1/2|$, where the probability is over the random choices made by the challenger and the adversary. We say that an identity-based encryption scheme is semantically secure against adaptive chosen ciphertext attack if no polynomially bounded adversary (in k) has non-negligible advantage (in k) in the game described above.

4 Secure Identity-Based Encryption Schemes

The cryptosystems described in this section make use of a *bilinear map* $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic groups of order p for some large prime p.[6] The map must satisfy the following properties:

1. Bilinear: For all $P, Q, R, S \in \mathbb{G}_1$, $\hat{e}(P+Q, R+S) = \hat{e}(P,R)\hat{e}(P,S)\hat{e}(Q,R)\hat{e}(Q,S)$.[7]

2. Non-Degenerate: For a given point $Q \in \mathbb{G}_1$, $\hat{e}(Q,R) = 1_{\mathbb{G}_2}$ for all $R \in \mathbb{G}_1$ if and only if $Q = 0_{\mathbb{G}_1}$[8]. From that and bilinearity, we can find that if P is a generator of $\mathbb{G}_1$, then $\hat{e}(P,P)$ is a generator of $\mathbb{G}_2$.

3. Computable: There is an efficient algorithm to compute $\hat{e}(P,Q)$ for any $P, Q \in \mathbb{G}_1$.

The security of the schemes in this section is based on the *Bilinear Diffie-Hellman Problem* (BDHP). The BDHP in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic groups of order p and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a bilinear map, can be stated as follows: *Given a generator P of $\mathbb{G}_1$ and three elements $aP, bP, cP \in \mathbb{G}_1$ for a, b, c random in $\mathbb{Z}_p$, compute $\hat{e}(P,P)^{abc}$.*

The Weil and Tate pairings on elliptic curves are the only known ways to build secure bilinear maps ([21]). We refer the reader to [15] section 6 or [21] for more details on these constructions.

4.1 The Boneh-Franklin Scheme

The first efficient and secure identity-based encryption scheme was given by Boneh and Franklin in [2]. To encrypt a message, the sender uses the bilinear map to combine the identity of the receiver, the PKG's public key and a random short term private key into a session key used to mask the message. The receiver can recreate the same session key by using the bilinear map to combine his private key and the short term public key sent with the ciphertext. Here is the description of the scheme in full details:

Setup: Given a security parameter k,
(1) generate cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ of prime order p together with a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ corresponding to this security parameter (say p could be a k-bit prime). Pick a random generator $P \in \mathbb{G}_1$.
(2) pick a random $s \in \mathbb{Z}_p^*$ and compute $P_{pub} = sP$.
(3) pick cryptographic hash functions
$H_1 : \{0,1\}^* \to \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2 \to \{0,1\}^n$,
$H_3 : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_p^*$,
$H_4 : \{0,1\}^n \to \{0,1\}^n$ for some integer $n > 0$.
The plaintext space is $\mathcal{M} = \{0,1\}^n$ and the ciphertext space is $\mathcal{C} = \mathbb{G}_1^* \times \{0,1\}^n \times \{0,1\}^n$. The public system parameters are $\mathsf{params} = \langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, p, n, P, P_{pub}, H_1, H_2, H_3, H_4 \rangle$. The master-key is s.

Extract: Given a string $\mathsf{ID} \in \{0,1\}^*$, the master-key s and system parameters params, compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}_1^*$ and $d_{\mathsf{ID}} = sQ_{\mathsf{ID}}$, and return d_{ID}.

[6] For consistency with previously published literature, we denote $\mathbb{G}_1$ additively and $\mathbb{G}_2$ multiplicatively.

[7] In particular, $\hat{e}(aP, bQ) = \hat{e}(P,Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p$

[8] This is the identity element in $\mathbb{G}_1$. We denote the identity element in $\mathbb{G}_2$ by $1_{\mathbb{G}_2}$

Encrypt: Given a plaintext $M \in \mathcal{M}$, a public key ID and public parameters params,
(1) compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID})$,
(2) pick a random $\sigma \in \{0,1\}^n$ and compute $r = H_3(\sigma, M)$,
(3) compute $g = \hat{e}(P_{pub}, Q_{\mathsf{ID}})$,
(4) set the ciphertext to $C = \langle rP, \sigma \oplus H_2(g^r), M \oplus H_4(\sigma)\rangle$.

Decrypt: Given a ciphertext $\langle U, V, W\rangle \in \mathcal{C}$, a private key d_{ID} and system parameters params,
(1) compute $g' = \hat{e}(U, d_{\mathsf{ID}})$,
(2) compute $\sigma = V \oplus H_2(g')$,
(3) compute $M = W \oplus H_4(\sigma)$,
(4) compute $r = H_3(\sigma, M)$. If $U \neq rP$, reject the ciphertext, else return M.

Note that M is encrypted as $W = M \oplus H_4(\sigma)$. This can be replaced by $W = E_{H_4(\sigma)}(M)$ where E is a semantically secure symmetric encryption scheme [9].

Consistency of this scheme easily follows from the bilinearity of $\hat{e}$. This scheme is semantically secure against adaptive chosen ciphertext attack in the random oracle model[10] if the BDHP is intractable in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}\rangle$. See [2] for a full proof of security and exact security analysis.

In this scheme, as in all other identity-based encryption schemes, the security of the PKG's master-key is capital because the security of all other private keys depend on it. One way to increase its security is to distribute the master-key among various sites using techniques of threshold cryptography [16]. The master-key is distributed in an t-out-of-n fashion by giving each PKG one share s_i if a Shamir secret sharing of $s \bmod q$. Given t PKG responses $Q^{(i)}_{priv} = s_i Q_{\mathsf{ID}}$, a user could then construct $d_{\mathsf{ID}} = \sum \lambda_i Q^{(i)}_{priv}$ where the λ_i's are the appropriate Lagrange coefficients.

[9] See [2] and [14] for more details.

[10] This means that in the security proof, the hash functions are modeled by random functions which are accessible to both the challenger and the adversary.

4.2 Authenticated ID-Based Encryption

Lynn [22] found that the Boneh-Franklin scheme could be modified to provide message authentication at no additional computational cost, i.e. upon reception, the receiver can verify the identity of the sender and whether or not the message has been tampered with. This eliminates the need for digital signatures when authentication is required. The level of security achieved is the same as that in a private conversation, i.e. secure authenticated communication without the ability to prove to a third party that any information was ever exchanged. To provide origin authentication, the bilinear map is now used to combine the identity of both the sender and the receiver. The session key is hashed with a random value to obtain a different mask each time the encryption function is executed.

Setup: Same steps as in the Boneh-Franklin scheme except that the hash function H_2 now must be defined as follows: $H_2 : \mathbb{Z}_p \times \mathbb{G}_2 \to \{0,1\}^n$ for some $n > 0$. The plaintext space is $\mathcal{M} = \{0,1\}^n$, the ciphertext space is $\mathcal{C} = \mathbb{Z}_p \times \{0,1\}^n \times \{0,1\}^n$. The public system parameters are params $= \langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, p, n, P, H_1, H_2, H_3, H_4\rangle$. The master-key is s.

Extract: Same as for the Boneh-Franklin scheme.

Encrypt: Given a plaintext $M \in \mathcal{M}$, a private key d_{ID_A}, a public key ID_B and system parameters params,
(1) pick a random $\sigma \in \{0,1\}^n$,
(2) compute $r = H_3(\sigma, M)$,
(3) compute $g = \hat{e}(d_{\mathsf{ID}_A}, H_1(\mathsf{ID}_B))$,
(4) set the ciphertext to $C = \langle r, \sigma \oplus H_2(r, g), M \oplus H_4(\sigma)\rangle$.

Decrypt: Given a ciphertext $\langle U, V, W\rangle \in \mathcal{C}$, a public key ID_A, a private key d_{ID_B} and system parameters params,
(1) compute $g = \hat{e}(H_1(\mathsf{ID}_A), d_{\mathsf{ID}_B})$,
(2) compute $\sigma = V \oplus H_2(U, g)$,
(3) compute $M = W \oplus H_4(\sigma)$,

(4) compute $r = H_3(\sigma, M)$. If $U \neq r$, reject the ciphertext, else return M.

Again, $W = M \oplus H_4(\sigma)$ can be replaced by $W = E_{H_4(\sigma)}(M)$ where E is a semantically secure symmetric cryptosystem.

Consistency of this scheme easily follows from the bilinearity of $\hat{e}$. Lynn shows that the scheme is secure against adaptive chosen ciphertext attack in the random oracle model provided that the BDHP is intractable in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$.[11] Under the same assumptions, he shows that an adversary has only negligible probability of forging a valid ciphertext from a sender ID_A to a receiver ID_B, even if he is given access to extraction, encryption and decryption oracles, provided he has not queried the private keys corresponding to ID_A or ID_B. Therefore the ciphertext is its own authentication code, i.e. if a ciphertext is valid, then it is authentic. See [22] for full proofs of security and exact security analysis.

We mention that in [23], Malone-Lee gives an identity-based signcryption scheme based on the BDHP, i.e. a scheme in which the ciphertexts are authenticated and non-repudiable[12]. However, he does not give any formal proof of security.

4.3 Hierarchical ID-Based Encryption

One disadvantage of the two previous schemes is that, in a large network, the private key generator would have a quite burdensome job. One solution to this problem is to allow a hierarchy of PKGs in which the PKGs have to compute private keys only to the entities immediately below them in the hierarchy (see Figure 2). In such a system, the users are no longer represented by a string ID, but by a tuple of IDs containing the ID of each of their 'ancestors' in the hierarchy. For example, $\langle \mathsf{ID}_1, \ldots, \mathsf{ID}_i \rangle$ would be the parent of $\langle \mathsf{ID}_1, \ldots, \mathsf{ID}_{i+1} \rangle$. We present a scheme by Gentry and Silverberg [17], which extends the Boneh-Franklin scheme to obtain a fully scalable hierarchical ID-based encryption scheme (the two schemes are identical if there is only one level). We note that in this scheme, the ID tuple of any entity in the hierarchy, except the root, can be used as a public key. To simplify the notation, we write $\overline{\mathsf{ID}}_{(i)}$ to denote $\langle \mathsf{ID}_1, \ldots, \mathsf{ID}_i \rangle$

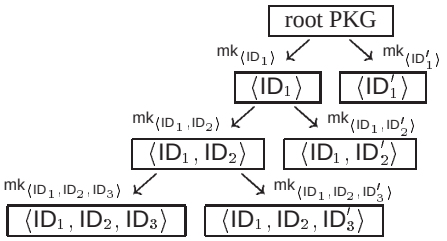

Figure 2: Hierarchy of PKG's

Root Setup: Given a security parameter k
(1) generate cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ of prime order p together with a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ corresponding to this security parameter (say p is a k-bit prime). Pick a random generator $P_\epsilon \in \mathbb{G}_1$.
(2) pick a random $s_\epsilon \in \mathbb{Z}_p^*$ and compute $Q_\epsilon = s_\epsilon P_\epsilon$.
(3) pick cryptographic hash functions
$H_1 : \{0,1\}^* \to \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2^* \to \{0,1\}^*$,
$H_3 : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_p^*$ and
$H_4 : \{0,1\}^n \to \{0,1\}^n$ for some integer $n > 0$.
The plaintext space is $\mathcal{M} = \{0,1\}^n$ and the ciphertext space is $\mathcal{C} = \mathbb{G}_1^{*\,l} \times \{0,1\}^n \times \{0,1\}^n$. The public system parameters are $\mathsf{params} = \langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, p, n, P_\epsilon, Q_\epsilon, H_1, H_2, H_3, H_4 \rangle$. The root secret is s_ϵ.

Lower-level Setup: Given the system parameters params, each entity $E_{\overline{\mathsf{ID}}_{(i)}}$ other than the root picks a random $s_{\overline{\mathsf{ID}}_{(i)}} \in \mathbb{Z}_p^*$ and computes $Q_{\overline{\mathsf{ID}}_{(i)}} = s_{\overline{\mathsf{ID}}_{(i)}} P_\epsilon$, which it keeps secret.

[11] The security model from Section 3 must be slightly modified to account for the fact that the encryption function now requires a private key, but the idea is the same.

[12] In the scheme by Lynn, the ciphertexts are repudiable since there is no difference between a ciphertext encrypted by A and sent to B and a ciphertext encrypted by B and sent to A.

Extract: Given the ID tuple $\langle \mathsf{ID}_1, \ldots, \mathsf{ID}_i \rangle$ of one of its children, its private key $\mathsf{mk}_{\overline{\mathsf{ID}}_{(i-1)}} = \langle S_{\overline{\mathsf{ID}}_{(i-1)}}, Q_\epsilon, Q_{\overline{\mathsf{ID}}_{(1)}}, \ldots, Q_{\overline{\mathsf{ID}}_{(i-1)}} \rangle$, its secret value $s_{\overline{\mathsf{ID}}_{(i-1)}}$[13] and system parameters params, PKG $E_{\overline{\mathsf{ID}}_{(i-1)}}$ computes the private key as follows:
(1) compute $P_{\overline{\mathsf{ID}}_{(i)}} = H_1(\overline{\mathsf{ID}}_{(i)})$,
(2) compute $S_{\overline{\mathsf{ID}}_{(i)}} = S_{\overline{\mathsf{ID}}_{(i-1)}} + s_{\overline{\mathsf{ID}}_{(i-1)}} P_{\overline{\mathsf{ID}}_{(i)}}$,
(3) return $\langle S_{\overline{\mathsf{ID}}_{(i)}}, Q_\epsilon, Q_{\overline{\mathsf{ID}}_{(1)}}, \ldots, Q_{\overline{\mathsf{ID}}_{(i-1)}} \rangle$.
The private key corresponding to $\langle \mathsf{ID}_1, \ldots, \mathsf{ID}_i \rangle$ is $\langle S_{\overline{\mathsf{ID}}_{(i)}}, Q_\epsilon, Q_{\overline{\mathsf{ID}}_{(1)}}, \ldots, Q_{\overline{\mathsf{ID}}_{(i)}} \rangle$ (the user $\langle \mathsf{ID}_1, \ldots, \mathsf{ID}_i \rangle$ already knows $Q_{\overline{\mathsf{ID}}_{(i)}}$).

Encrypt: Given a plaintext $M \in \mathcal{M}$, an ID tuple $\langle \mathsf{ID}_1, \ldots, \mathsf{ID}_l \rangle$ and system parameters params,
(1) compute $P_{\overline{\mathsf{ID}}_{(i)}} = H_1(\overline{\mathsf{ID}}_{(i)})$ for $1 \leq i \leq l$,
(2) compute $g = \hat{e}(Q_\epsilon, P_{\overline{\mathsf{ID}}_{(1)}})$,
(3) pick a random $\sigma \in \{0,1\}^n$ and compute $r = H_3(\sigma, M)$,
(4) set the ciphertext to $C = \langle rP_\epsilon, rP_{\overline{\mathsf{ID}}_{(2)}}, \ldots, rP_{\overline{\mathsf{ID}}_{(l)}}, \sigma \oplus H_2(g^r), M \oplus H_4(\sigma) \rangle$.

Decrypt: Given a ciphertext $C = \langle U_0, U_2, \ldots, U_l, V, W \rangle \in \mathcal{C}$, a private key $\langle S_{\overline{\mathsf{ID}}_{(l)}}, Q_\epsilon, Q_{\overline{\mathsf{ID}}_{(1)}}, \ldots, Q_{\overline{\mathsf{ID}}_{(l)}} \rangle$ and system parameters params,
(1) compute

$$g' = \frac{\hat{e}(U_0, S_{\overline{\mathsf{ID}}_{(l)}})}{\prod_{i=2}^{l} \hat{e}(Q_{\overline{\mathsf{ID}}_{(i-1)}}, U_i)},$$

(2) compute $\sigma = V \oplus H_2(g')$,
(3) compute $M = W \oplus H_4(\sigma)$,
(4) compute $r = H_3(\sigma, M)$. If $U_0 \neq rP_\epsilon$, reject the ciphertext, else return M.

As before, we can replace $W = M \oplus H_4(\sigma)$ by $W = E_{H_4(\sigma)}(M)$ where E is a semantically secure symmetric cryptosystem.

For this setting, we modify our security model so that the adversary is not allowed to posses the private key of any ancestor of the entity on which he is being challenged. Gentry and Silverberg proved that the scheme is secure against adaptive chosen ciphertext attack in the random oracle model provided that the BDHP is intractable in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$.

We mention that Horwitz and Lynn [19] also proposed a 2-level hierarchical ID-based encryption scheme, but it has only limited resistance to user collusion.

5 Summary and Open Problems

We surveyed recent proposals for usable identity-based encryption schemes. The schemes based on bilinear maps seem most promising because we have precise proofs of their security in a strong security model. However, the security of these schemes is based on a new hardness assumption which has not been studied much. It would be interesting to see if we can relate the Bilinear Diffie-Hellman Problem to other well studied problems (Cheon and Lee [6] and Yacobi [32] made first steps in this direction).

As for Cocks' scheme, a decrease in the expansion factor would also be a major improvement to the scheme. One could also try to relate the Quadratic Residuosity Problem to the Factoring Problem.

It is also an open problem to design an identity-based encryption scheme that is secure in the standard computational model rather than in the random oracle model. Boneh and Franklin [2] mention that one might hope to do this by modifying the Boneh-Franklin scheme using the techniques of Cramer-Shoup [8] to obtain a scheme based on the decision analog of the BDHP. Another interesting problem would be to build identity-based encryption schemes based on other complexity assumptions.

We mention that many results have been published

[13] For definiteness, if $i = 1$, then the private key is $\mathsf{mk}_\epsilon = \langle S_\epsilon, Q_\epsilon \rangle$ where S_ϵ is the identity element in $\mathbb{G}_1$, and PKG's secret value is s_ϵ.

in other areas of identity-based cryptography, namely, Paterson [25], Hess [18] and Cha and Cheon [3] each proposed new identity-based signature schemes, and Smart [29], Zhang, Liu and Kim [33] and Chen and Kudla [5] each proposed new identity-based key agreement protocols.

Acknowledgments

I would like to thank Matthew Franklin and Markus Jakobsson for their comments on previous versions of this paper.

References

[1] D. Boneh, X. Ding and G. Tsudik. *Identity based encryption using mediated RSA*, in 3rd Workshop on Information Security Application, Jeju Island, Korea, Aug. 2002.

[2] D. Boneh and M. Franklin. *Identity Based Encryption from the Weil Pairing*, to appear in SIAM Journal of Computing.

[3] J. Cha and J. Cheon. *An Identity-Based Signature from Gap Diffie-Hellman Groups*, PKC 2003, LNCS 2567 pp. 18-30, 2002.

[4] R. Canetti, S. Halevi and J. Katz. *A Forward-Secure Public-Key Encryption Scheme*, to appear in Proceedings of Eurocrypt 2003.

[5] L. Chen and C. Kudla. *Identity Based Authenticated Key Agreement from Pairings*, Cryptology ePrint Archive[14], Report 2002/184, 2002.

[6] J. Cheon and D. Lee. *Diffie-Hellman Problems and Bilinear Maps*, Cryptology ePrint Archive, Report 2002/117, 2002.

[14] http://eprint.iacr.org/

[7] C. Cocks. *An Identity Based Encryption Scheme Based on Quadratic Residues*, Cryptography and Coding, LNCS 2260, pp. 360-363, 2001.

[8] R. Cramer and V. Shoup. *A practical public key cryptosystem probably secure against adaptive chosen ciphertext attack*, Proceedings of Crypto '98, LNCS 1462, pp. 13-25, 1998.

[9] Y. Desmedt and J.-J. Quisquater. *Public-key systems based on the difficulty of tampering*, Proceedings of Crypto '86, LNCS 263, pp. 111-117, 1986.

[10] G. Di Crescenzo, R. Ostrovsky and S. Rajagopalan. *Conditional Oblivious Transfer and Timed-Release Encryption*, Proceedings of Eurocrypt '99, LNCS 1592, pp. 74-89, 1999.

[11] Y. Dodis, M. Franklin, J. Katz, A. Miyaji and M. Yung. *Intrusion-Resilient Public-Key Encryption*, to appear in RSA 2003 – Cryptographer's Track.

[12] U. Feige, A. Fiat and A. Shamir. *Zero-Knowledge Proofs of Identity*, Journal of Cryptology, Vol. 1, pp. 77-94, 1988.

[13] A. Fiat and A. Shamir. *How to Prove Yourself: Practical Solutions to Identification and Signature Problems*, Proceedings of Crypto '86, LNCS 263, pp. 186-194, 1987.

[14] E. Fujisaki and T. Okamoto. *Secure Integration of Asymmetric and Symmetric Encryption Schemes*, Proceedings of Crypto '99, pp. 537-554, 1999.

[15] M. Gagné. *Applications of Bilinear Maps in Cryptography*, Master's thesis, University of Waterloo, 2002. available at http://wwwcsif.cs.ucdavis.edu/~gagne/thesis.pdf

[16] P. Gemmel. *An Intoduction to Threshold Cryptography*, CryptoBytes, a technical newsletter of RSA Laboratories, Vol. 2, No. 7, 1997.

[17] C. Gentry and A. Silverberg. *Hierarchical ID-Based Cryptography*, Proceedings of Asiacrypt 2002, LNCS 2501 pp. 548-566, 2002.

[18] F. Hess. *Exponent Group Signature Schemes and Efficient Identity Based Signature Schemes Based on Pairings*, Cryptology ePrint Archive, Report 2002/012, 2002.

[19] J. Horwitz and Ben Lynn. *Toward Hierarchical Identity-Based Encryption*, Proceedings of Eurocrypt 2002, LNCS 2332, pp. 466-481, 2002.

[20] D. Hühnlein, M. Jacobson and D. Weber. *Towards Practical Non-interactive Public Key Cryptosystems Using Non-maximal Imaginary Quadratic Orders*, Proceedings of SAC 2000, LNCS 2021, pp. 275-287, 2000.

[21] A. Joux. *The Weil and Tate Pairings as Building Blocks for Public Key Cryptosystems*, ANTS 2002, LNCS 2369, pp. 20-32, 2002.

[22] B. Lynn. *Authenticated Identity-Based Encryption*, Cryptology ePrint Archive, Report 2002/072, 2002.

[23] J. Malone-Lee. *Identity-Based Signcryption*, Cryptology ePrint Archive, Report 2002/098, 2002.

[24] U. Maurer and Y. Yacobi. *Non-interactive public-key cryptosystem*, Proceedings of Eurocrypt '91, LNCS 547, pp. 498-507, 1991.

[25] K. Paterson. *ID-Based Signatures from Pairings on Elliptic Curves*, Electronics Letters, Vol. 38 (18), 1025-1026, 2002.

[26] C. Rackoff and D. Simon. *Noninteractive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack*, Proceedings of Crypto '91, pp. 433-444, 1991.

[27] R. Rivest, A. Shamir and D. Wagner. *Time lock puzzles and timed release cryptography*, Technical report, MIT/LCS/TR-684.

[28] A. Shamir. *Identity-Based Cryptosystems and Signature Schemes*, Proceedings of Crypto '84, pp. 47-53, 1984.

[29] N. Smart. *An Identity Based Authenticated Key Agreement Protocol Based on the Weil Pairing*, Electronics Letters, Vol 38, pp 630-632, 2002.

[30] S. Tsuji and T. Itoh. *An ID-based cryptosystem based on the discrete logarithm problem*, IEEE Journal on Selected Areas in Communication, vol. 7, no. 4, pp. 467-473, 1989.

[31] H. Tanaka. *A realization scheme for the identity-based cryptosystem*, Proceedings of Crypto '87, LNCS 293, pp. 341-349, 1987.

[32] Y. Yacobi. *A Note on the Bilinear Diffie-Hellman Assumption*, Cryptology ePrint Archive, Report 2002/113, 2002.

[33] F. Zhang, S. Liu and K. Kim. *ID-Based One Round Authenticated Tripartite Key Agreement Protocol with Pairings*, Cryptology ePrint Archive, Report 2002/122, 2002.

Advances in Side–Channel Cryptanalysis

Electromagnetic Analysis and Template Attacks

Dakshi Agrawal Bruce Archambeault Suresh Chari Josyula R. Rao
Pankaj Rohatgi
IBM T. J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598
email: {agrawal,barch,schari,jrrao,rohatgi}@us.ibm.com

Abstract

We describe two recent advances which substantially increase the scope and power of side–channel cryptanalysis. The first advance is the exploitation of information leakage from electromagnetic emanations. The second advance, known as template attacks, is a superior data analysis technique which substantially reduces the number of side–channel samples needed for an attack. These advances pose a risk to all cryptographic implementations, including those immune against earlier side–channel attacks.

1 Introduction

Side–channel cryptanalysis has emerged as an extremely powerful and practical tool for breaking commercial implementations of cryptography. These attacks exploit the fact that implementations of cryptography on physical devices leak much more information than just the input–output relationship [12, 13]. A large number of attacks have been published, exploiting leakages from subtle channels, such as the timing of operations and the instantaneous power consumption of the device.

In contrast to most cryptanalytic attacks, timing and power attacks are extremely easy to mount, while their consequences are equally devastating. Hence, vendors expend substantial effort to protect their products against such attacks. While timing attacks are widely applicable, simple countermeasures are quite effective and have been incorporated in most implementations. On the other hand, power–analysis attacks are notoriously hard to counter: Even though sound countermeasures are available [9, 4], applying them is an error-prone task. Fortunately, such attacks only afflict smart cards and other simple devices where the unfiltered power line is readily accessible. As a consequence, while third–party validation of power–analysis countermeasures has become the norm for smart card development, vendors of devices with inaccessible power lines have been spared. Some vendors have also attempted to side–step these attacks by throwing in ad–hoc countermeasures at a protocol level. For example, implementations vulnerable to statistical attacks on multiple invocations are protected by protocol–tweaks which limit the number of invocations available to the attacker.

In this article, we describe two recent advances which substantially increase the scope and power of side–channel attacks. These advances pose a significant risk to *all* existing cryptographic implementa-

tions, including those deemed secure against power–analysis attacks. Implementations relying on inaccessible power lines or ad–hoc, protocol–level countermeasures are especially vulnerable to these advances.

The first advance is the practical exploitation of information leaked from electromagnetic (EM) emanations [1]. For a long time it was rumored that EM–based attacks were an extremely powerful tool for espionage. It is known that defense organizations go to great lengths to contain EM emanations from their equipment and facilities. Not surprisingly, large amounts of information about EM–analysis continues to be classified [18, 10]. Our work in [1] shows that these rumors and precautions were fully justified: Not only does EM–analysis provide an avenue for attacking cryptographic devices from a distance where the power line is inaccessible, it also provides information not available in the power side–channel.

The second advance, termed *template attacks* in [5], is a technique for attacking cryptographic implementations where an attacker is limited to very few invocations, and where traditional side channel attacks do not work. Such situations arise frequently in the case of stream ciphers and in cases where protocol–level countermeasures to side–channel analysis have been deployed. The technique is motivated by results from signal detection and estimation theory. It uses a test device, identical to the target device being attacked, to build detailed noise models of the side–channel for different device states. These models (or *templates*) are used to classify the available signal(s) from the target device. The sophistication and accuracy of these models determines how close the technique approaches optimality in terms of utilizing all the information present within the available signals.

While our advances substantially improve the state of the art in side–channel attacks, defending against them need not be more onerous than defending against power–analysis attacks. Sound randomization–based countermeasures against power–analysis [4, 9] are applicable to all types of limited information leakage. Randomized implementations are usually immune against template attacks, since the adversary cannot force his test device to mimic the random choices made by the target device. To defend against EM attacks, hardware vendors must profile all EM leakages from their raw hardware and take steps to reduce egregious ones. Implementors of cryptography on such hardware should take into account the net leakage from the power and EM channels to select the appropriate countermeasures. We hope that, in light of these advances, implementors will treat side–channel attacks seriously and address them using sound countermeasures rather than ad–hoc ones.

2 EM Emanation Analysis

2.1 Understanding EM Emanations

There are two broad categories of EM emanations:

1. Direct Emanations: These result from *intentional* current flows within circuits. Many of these consist of short bursts of current with sharp rising edges, resulting in emanations over a wide frequency band. Often, components at higher frequencies are more useful to the attacker due to noise and interference prevalent in the lower bands. In complex circuits, isolating direct emanations can be very difficult due to interference from other signals. Minimizing interference requires the use of tiny field probes positioned very close to the signal source and/or special filters to extract the desired signal.

2. Unintentional Emanations: Increased miniaturization and complexity of modern CMOS devices results in electrical and electromagnetic coupling between components in close proximity. Small couplings, typically ignored by circuit designers, provide a rich source of compromising emanations. These emanations manifest themselves as *modulations* of the carrier signals generated, present or introduced within

the device. Typical sources of such carriers include the harmonic–rich clock signal(s) and signals used for internal and external communication. Depending on the type of coupling, the carrier can be either *Amplitude Modulated* (AM) or *Angle Modulated* (e.g., FM) by the sensitive signal, or the modulation could be more complex. If the modulated carrier can be captured, then the sensitive signal can be recovered using an EM receiver tuned to the carrier frequency and performing the appropriate demodulation.

Initial published work on EM–analysis [11, 16] focused exclusively on direct emanations. However, the resulting attacks were limited in scope. The best attacks were semi-invasive and required careful positioning of micro-antennas on the passivation layer of the chip substrate. Such attacks were not known to be better than power–analysis attacks.

The key to unlocking the power of the EM side–channel lies in exploiting unintentional emanations rather than direct emanations. Some modulated carriers are much stronger and propagate much further than direct emanations. This enables attacks to be carried out at a distance without resorting to any invasive techniques. This situation has an analogue in astronomy where planets around distant stars are detected not via direct observation but via indirect observation. While the reflected light coming from a planet is quite feeble and easily overwhelmed by light from the star, a planet affects the star's light in measurable ways. For example, a revolving planet can produce a noticeable wobble in the star's position due to gravitational effects, or produce a detectable dimming of the star's light when it comes between the star and the Earth.

2.2 EM Attack Equipment

Just like power–analysis, an EM attack system requires sample collection equipment such as a digital oscilloscope or a PC-based data sampling card. The *critical* piece of equipment for enabling EM attacks is an EM receiver/demodulator which can be tuned to various modulated carriers and performs demodulation to extract the sensitive signal. Here, the main tradeoff is between cost and convenience. Those with budgets in the tens of thousands of dollars can buy a new/used high-end receiver such as the Dynamic Sciences R-1550 [7], which covers a wide band and offers the user a large selection of bandwidths and demodulation options. Those on low budgets could settle for a used ICOM IC-R7000 receiver which can be had for under $1000, but provides only limited bandwidth, is noisier, and requires software to demodulate its IF output. Those on low budgets and unwilling to put up with noise and limited bandwidth can construct their own receiver for under $1000 by using commonly available low–noise electronic components and demodulation software; the additional inconvenience with this approach is the need for frequent calibration. Picking up EM signals also requires the use of EM near-field probes and antennas appropriate for the band being considered. However, these items are not expensive and can even be assembled using low-cost materials from a hardware store.

2.3 EM Attacks on an RSA–Accelerator

We illustrate the power of the EM side-channel by means of an example of special interest to the readers of this newsletter. We analyzed a commercial, PCI–based SSL/RSA accelerator, R [1], installed in an Intel-based server. We programmed the server to repeatedly decrypt a fixed ciphertext, *i.e.* invoke R to perform modular exponentiation with the given ciphertext, modulus and exponent.

In a real–world setting, mounting power and timing attacks on such a setup is infeasible and/or risky. The server's power supply is well filtered and contaminated by large currents from a number of components. For power–analysis, the attacker will have to physically open the server to access R, and make addi-

[1] We are using a pseudonym to protect vendor identity. R is rated to perform 200, 1024-bit CRT based RSA private key ops/s.

tional hardware modifications to tap the power/ground lines feeding the RSA engine. Theoretically, R suffers from a timing attack, due to conditional subtraction in the Montgomery reduction step [15, 6]. With perfect timing information, this attack can extract reasonably sized keys (1024-bits or more) with a few million timing samples. However, in practice, the timing obtained by interacting with the server will be very inaccurate due to random delays introduced by network latency, server load, server OS, server-to-R communication, etc. Compensating for this inaccuracy will require a several–fold increase in the number of timing samples, thus rendering this attack infeasible.

The situation changes dramatically when EM emanations are considered. Even though R is inside a closed server, a large number of carriers are available on the outside. This holds not just for R but for *all* RSA accelerators that we have tested, including some designed to meet the tamper resistance standard of FIPS. One typically finds many high energy carriers at multiples of the accelerator's clock frequency and several intermediate strength *intermodulated* carriers at other frequencies. These intermodulated carriers arise due to non-linear interactions among the various carriers present within the accelerator's operating environment. The presence of so many signals permits a variety of attacks to be mounted at various distances from the device.

Even at distances of fifty feet and through walls and glass, one can capture some high energy emanations from R. These are mostly modulated carriers at the odd–harmonics of R's internal clock. AM–demodulating these carriers yields signals where the start and end of the modular exponentiation is clearly delimited. These signals greatly enhance the timing attack on R, to the point where it may even become practical. An EM detector stationed inconspicuously in another room, 40-50 feet away, could precisely measure RSA operation timing, regardless of network/server/communication latency. Such an EM-enhanced timing attack still has a few disadvantages:

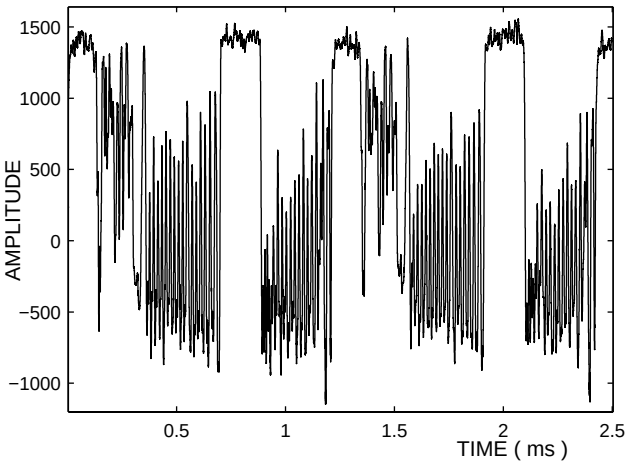

Figure 1: EM Signal from SSL Accelerator R.

Firstly, a large number of invocations are needed, and secondly, the attack does not work if the software controlling R implements the RSA data–blinding countermeasure[12].

If, on the other hand, the adversary can get a small EM capturing/retransmitting device within 4-5 feet from R, he will have a wide range of EM attacks at his disposal, including attacks which use a single invocation and attacks which bypass the data blinding defense. At the very least, he can launch EM versions of the numerous published power–analysis attacks that exploit specific leakages that occur in RSA hardware found in smart cards. Unless R has been specifically designed to resist all known power–analysis attacks, there is no hope that it can survive EM attacks at this distance. The reader will better appreciate this situation by looking at the quality of information about RSA internals that is available in these EM emanations: It is at least as good, if not better, than what is available in power–analysis scenarios.

Figure 1 shows the signal obtained by AM demodulating a 461.46Mhz intermodulated carrier with a band of 150Khz for a period of 2.5ms during which R com-

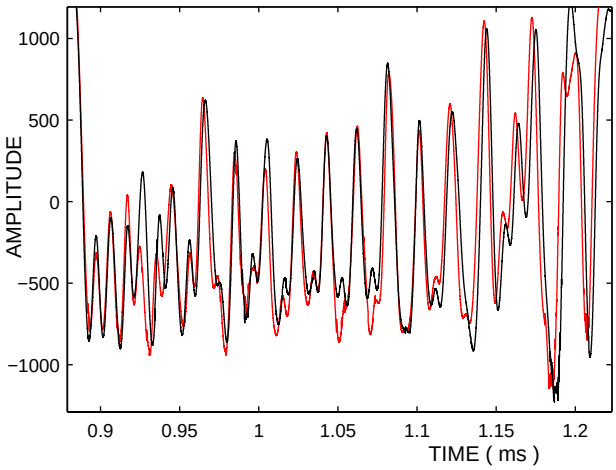

Figure 2: Same exponent, different data.

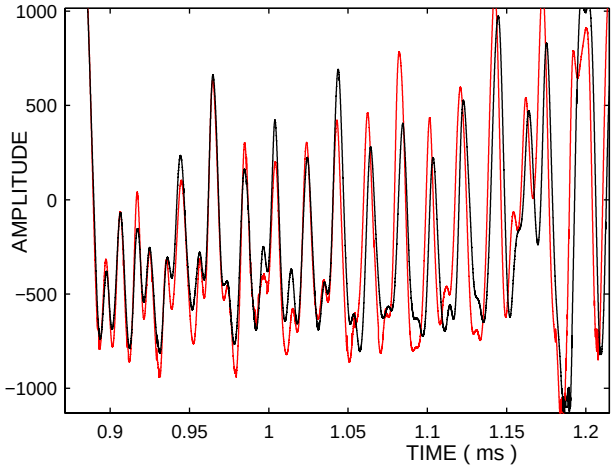

Figure 3: Same data, different exponents.

putes two successive and identical 2048-bit modular exponentiations with a 12-bit exponent. For clarity, the figure shows an average taken over ten signal samples. One can clearly see a basic signal shape repeated twice, with each repetition corresponding to a modular exponentiation. The first repetition spans the time interval from 0 to 1.2ms and the second from 1.2ms to 2.4ms. The signal also shows the internal structure of the exponentiation operation. From time 0ms to 0.9ms, R receives the exponentiation request and performs some precomputation to initialize itself to exponentiate using the Montgomery method. The actual 12-bit exponentiation takes place approximately from time 0.9ms to 1.2ms. A closer inspection of this region reveals substantial information leakage which is beneficial to an adversary. Figure 2 plots an expanded view of this region for two different exponentiation requests which have the same modulus and exponent but different data. The two signals are plotted in different colors. From the start, one can see that the two signals go in and out of alignment due to data dependent timing of the Montgomery multiplications employed by this implementation.

In fact, if an adversary knows the modulus and the data being exponentiated then, according to the work of [20], Montgomery multiplication timings for one or a few invocations are enough to recover the secret exponent. An earlier approach that also works is the MESD (multi-exponent, single data) style attack of [14]. In MESD, the adversary tries to match the observed signal with a signal generated from an identical RSA device on identical data by adjusting the bits of a prefix of a trial exponent. When the prefix of the bits of the trial exponent match those of the unknown exponent, the corresponding prefix of the observed signal matches the generated one. For example Figure 3 plots the region of exponentiation for two different exponentiation requests having the same modulus and data but different exponents. The first 5 bits of the two exponents are the same and, as can be seen, the two signals are initially aligned and timing differences only arise at around 1.05ms which is somewhere in the middle of the 12-bit exponentiation.

Even if the adversary does not know the modulus and the data being exponentiated (e.g., Chinese Remaindering and/or data blinding are used), the RSA implementation can still be broken using results of [3, 20, 17]. This is because the statistics of the conditional subtract operation in Montgomery reduction depend on whether a square or a multiply is carried out.

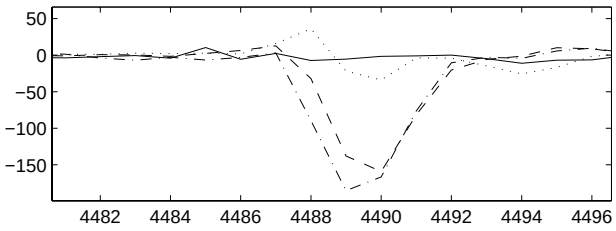

Figure 4: Leakage of an S-box output bit in 4 side–channels during Region 1 of computation. The solid line shows power channel leakage and different broken-line styles show leakages from 3 EM channels.

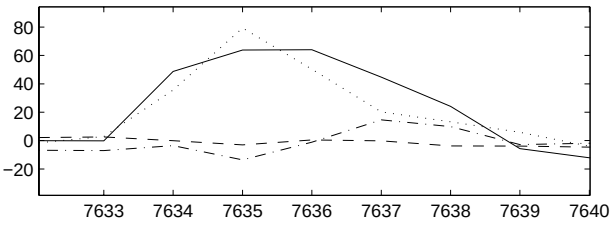

Figure 5: S-box bit leakage in power and EM channels in Region 2.

In fact, such timing statistics from just a few hundred traces are sufficient to recover the secret exponent.

At intermediate distances of 10-15 feet, the level of noise increases but there is still enough information to enable several attacks. In particular, the attack based on conditional–subtract statistics [3, 20, 17] still works on all RSA implementations, albeit with an increased number of, say, a few thousand samples. If an attacker at this distance is further limited to only a few samples, he could still mount template attacks on non-blinded, non-CRT RSA implementations to recover the key.

2.4 Multiplicity of EM channels

As mentioned earlier, several EM signals can be isolated from any device. This raises several interesting questions: One question is whether such a multiplicity is beneficial to an attacker, i.e., do different EM signals provide different types of information, or is there just one type of information leakage which happens to be present with differing magnitudes within different EM signals? Another important question is whether the EM side–channels as a whole are any better than the power side–channel, i.e., are EM–attacks still useful when the power side–channel is available?

We answer these questions by applying Differential Power Analysis (DPA) and Differential Electromagnetic Analysis (DEMA) attacks to measure information leakage. It is well known that the sensitive data (for example, a bit of the output of an S-box) has large co-variance with the side–channel signal at times when it leaks and zero co-variance when it does not. This is the underlying basis for attacks such as DPA and DEMA. Thus, by comparing the co-variance plots for the same sensitive data using different signals, we can compare its leakage across these different signals. We performed one DPA and three DEMA attacks on a DES implementation on a smart-card, based on predicting an S-box output bit in the first round. The DEMA attacks used different EM signals.

We plotted all four co-variance plots together and aligned them in time. The first thing we noted is that the most prominent leakages occur at different times in the different signals. This means that the mechanism of leakage is different in these signals. Figures 4 and 5 show two regions in time where some of the prominent leakages occur. These plots show the leakages of the S-box bit (i.e., the value of the co-variance) for all four signals with respect to time (in 10ns units). Each different signal is plotted in a different line–style, with the power leakage being a solid line and the 3 EM leakages plotted in different broken–line styles. In Figure 4 we see a region where there is substantial leakage in two EM channels, minor leakage in the third and no leakage in the power channel. This shows that EM channels have leakages

missed by the power channel. Figure 5 shows a region where the two EM channels which had large leakages in the earlier region failed to pick up a leakage that was picked up very well by the third EM channel as well as the power channel.

Now that we have strong evidence that different EM channels carry different information and that some EM channels leak certain information much better than the power channel, the true power of EM–analysis finally emerges. Not only can an attacker mount EM attacks in situations where the power channel is unavailable, sometimes he can use a single EM channel to mount attacks which are impractical using the power channel. This has serious implications for the security of current power–analysis resistant implementations. We have verified on smart cards that certain DPA resistant implementations are indeed vulnerable to single channel EM attacks.

In fact, the situation is much worse. An attacker could potentially do much better by combining information leakages from multiple EM channels and/or the power channel. Developing countermeasures for such attacks requires a methodology to assess the net information leakage from all the EM signals (and the power signal) realistically available to an adversary. Dealing with multiple channels and assessing their net information leakage is quite complex and sometimes counter–intuitive and is outside the scope of this paper. The interested reader can find more about our work on these aspects of the EM side–channel(s) in [2].

3 Template Attacks

Most of the devastating side–channel attacks reported in the literature can be classified as either Simple Power Analysis (SPA) or Differential Power Analysis (DPA) [13]. In SPA, the key can be extracted from a single (or few) sample(s) due to substantial leakages when executing key–dependent code. For instance, if the implementation contains key–dependent branching, the side-channel will usually reveal which branches were executed, hence yielding the key. SPA is also possible on implementations which use instructions for which sensitive information is clearly visible over and above the noise inherent in the side-channel. For example, if an instruction conditionally sets a carry bit, there could be enough leakage in the side–channel to disclose the value of this bit. When there is no key-dependent branching and low leakage instructions are used, SPA is not possible and statistical techniques such as DPA are needed. DPA relies on the statistical analysis of a *large* number of side–channel samples of the cryptographic operation, invoked with the same secret key and possibly differing data. Intuitively, with a large number of samples, the random noise component can be substantially reduced by averaging, and the smaller leakage signals extracted. To date, all side-channel attacks, even those using EM emanations, are variants of the basic techniques of SPA and DPA.

In [5], we showed that SPA/DPA-based techniques are sub-optimal since they do not exploit *all* the information available in each side–channel sample. Consequently, some implementations believed to be immune to side–channel attacks simply because the adversary is limited to one or at most a few compromising samples, can in reality be broken using template attacks which extract more information from the samples. Many such implementations are found in applications where higher-level protocols limit the number of samples an adversary can collect. Such situations also arise naturally in the case of many stream ciphers.

Consider an implementation of the stream cipher RC4. While there are cryptanalytic results highlighting minor statistical weaknesses, there are no major statistical biases that can be easily exploited by side–channel attacks. To our knowledge, no successful side–channel attack on a *reasonably designed*[2] RC4

[2] IEEE 802.11 uses RC4 in a protocol which reuses significant portions of the secret key, thus making implementations vulnerable to DPA and indeed to cryptographic attacks

implementation has been reported. In a well designed system, the RC4 cipher will always be initialized with a fresh secret key. Initializing the 256-byte internal state of RC4 using the secret key is simple enough to be implemented using low leakage instructions, in a key–independent manner. Thus, simple attacks such as SPA are unlikely. After initialization, the only secret is the internal state. However, this state evolves very rapidly as the cipher outputs more bytes. This rapidly evolving secret state is outside the control of the adversary. This provides inherent immunity against statistical attacks such as DPA, since the adversary cannot freeze the active part of the state to collect multiple samples.

For RC4, the best that an adversary can hope for is to obtain a *single* sample of the side–channel leakage during the key initialization phase. As mentioned above, a good implementation of RC4 will not be vulnerable to SPA on such a sample. We verified this fact by coding an implementation of RC4 on a smart card. However, the same implementation is easily broken with a single sample using the *template attack* technique. The template attack technique can theoretically extract all possible information available in each sample and is hence the strongest form of side–channel attack possible in an information theoretic sense.

Template attacks require the adversary to possess a programmable device identical to the device being attacked. While such an assumption is limiting, it is practical in many cases and has been used before in other side–channel attacks [8, 14]. In the following subsection, we outline the theoretical underpinnings of the template attack and describe some heuristics for making the attacks practical.

3.1 Template Attack Technique

Assume we have a device performing one of $\mathcal{K}$ possible operation sequences: For example, these could be the executions of the same code for $\mathcal{K}$ different values of some key bits. An adversary who gets a sample S of the side–channel during this operation wishes to identify which of the $\mathcal{K}$ operation sequences is being executed or to significantly reduce the set of of possibilities. We call this the *sample classification problem*.

In signal processing, it is customary to model the observed sample as a combination of an intrinsic signal component generated by the operation and a noise component which is intrinsically generated or ambient. Whereas the signal component is the same for repeated invocations of the operation, the noise component is best modeled as a random sample drawn from a noise probability distribution. This distribution depends on several factors such as the type of operation being performed and the physical operating environment. It is well known [19] that the optimal approach for solving the sample classification problem is to use the maximum likelihood approach, *i.e.*, the best guess is to pick the operation such that the probability of the observed noise component in S is maximized. Computing this probability requires the adversary to precisely model both the intrinsic signal and the noise probability distribution for each of the $\mathcal{K}$ operations. We refer to such models for intrinsic signals and noise probability distributions as *templates*. Once each of the $\mathcal{K}$ templates for the different operations are available, the sample classification problem can be solved.

Translating this theoretical technique into an attack presents several practical problems. The first problem is to obtain templates without complete and detailed physical specifications of the device to be attacked or even full access to it. We get around this problem by using an experimental device, identical to one to be attacked, to build the templates that are needed. While no two devices are *truly* identical, devices which come from the same hardware revision are similar enough for this technique to work.

The second problem is that it is difficult to estimate the noise probability distribution, since the noise is a real valued function of time, and even band-limited noise needs to be modeled as a T-dimensional

vector of reals in a certain range.[3] Even assuming smoothness properties, estimating the probability density function over this huge domain becomes infeasible when T is large. Luckily, modeling the noise of physical systems is a well-studied problem in Signal Detection and Estimation Theory [19] and we can use any of the several accurate and computationally feasible noise models that are available. For example, for our RC4 attack, we found that the *Multivariate Gaussian Noise* model is sufficient, whereas simpler models based on univariate statistics gave poor results. Most of the effort in estimating the multivariate Gaussian noise distribution goes towards computing the pairwise noise correlations for the T points, which requires around $O(T^2)$ work.

The third problem is that, in cryptographic settings, the key–finding problem does not directly translate into a sample classification problem since the value of $\mathcal{K}$, the entire key space, is huge. Clearly, building a template for each possible cryptographic key is infeasible. The solution is to meld the basic sample classification approach with details of the cryptographic operation being attacked. The result is a process of *iterative classification* of signals from the signal processing viewpoint and an *extend-and-prune* strategy from the perspective of searching the key space. The adversary uses the experimental device to identify a small prefix S_0 of the sample S, depending only on a few unknown key bits K_0. Using the experimental device, he builds templates for S_0 with each possible value for K_0. Using these templates, he classifies S_0, *i.e.*, prunes the set of possibilities for the values of K_0 being used in S_0 to a very small number. Then, in the next iteration, a longer prefix S_1 of S involving additional key bits K_1 is considered. Each remaining possible value of K_0 is then extended by all possible values of K_1, and templates are constructed for these key values using the longer prefix. Again, the sample S is used to prune the set of possible values for both K_0 and K_1. This process is repeated with longer and longer prefixes of S until all the key bits are covered and a manageable size set of possibilities for the entire key remains. The actual key can then be identified from this set by testing with known input/outputs.

The success of this strategy critically depends on how effectively the pruning process reduces the combinatorial explosion in the extension process. In general, the extent of information leakage from an implementation on a particular device inherently places theoretical bounds on the success of the template attack; the best an adversary can do is to approach this theoretical bound by building extremely accurate templates. In the particular case of cryptographic algorithms implemented on CMOS devices, the chances of success are likely to be quite good due to the twin properties of *contamination* and *diffusion*. Contamination refers to key–dependent leakages which can be observed over multiple cycles in a section of computation. In CMOS devices, direct manipulation of the key bits makes them part of the device state and these state leakages can persist for several cycles. Additionally, other variables affected by the key, such as key dependent table indices and values, cause further contamination at other cycles. The extent of contamination determines the level of success in pruning candidates for fresh key bits introduced in the expansion phase. However, if two keys are almost the same, even with contamination, pruning at this stage may not be able to eliminate one of them. Diffusion is the well-known cryptographic property wherein small differences in key bits are increasingly magnified in subsequent portions of the computation. Even when certain candidates for key bits are not eliminated due to contamination effects, diffusion will ensure that closely spaced keys will be pruned rapidly at later stages.

[3] T is the number of sampling points or, more exactly, the number of points used for sample classification

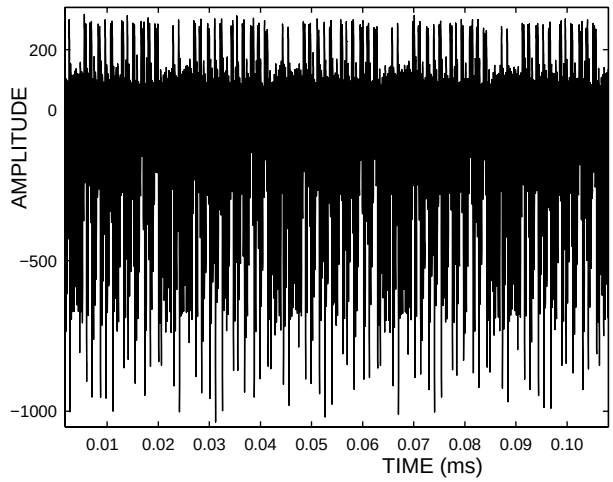

Figure 6: Power sample during first five iterations of RC4 state initialization loop.

3.2 Template attack on RC4

We describe how template attacks can be used on our implementation of RC4. RC4 operates on a 256-byte state table T to generate a pseudo-random stream of bytes that is then XOR'ed with the plaintext. Table T is initially fixed, and first, a variable length key (1 to 256 bytes) is used to update T using the pseudo code below:

```
i1 = i2 = 0;
for (ctr = 0; ctr < 256; ctr++)  {
  i2 = (key[i1] + T[ctr] + i2) % 256;
  swap_byte(&T[ctr], &T[i2]);
  i1 = (i1 + 1) % key_data_len;
}
```

A portion of the corresponding power side–channel is shown in Figure 6. The observable structural repetition is exactly five successive iterations of the loop. We first verified that simple side-channel analysis techniques do not work on our implementation. Figure 7 is based on side channel samples from the RC4 key initialization phase: The upper trace is the difference between two single power samples when the keys are the same, the lower trace when they are different. Contrary to expectation, the first case shows larger differences. This ambiguity exists even when one looks at differences of averages of up to five invocations, as shown in Figure 8. Clear and consistent differences emerge only when one considers averages of several tens of samples. Therefore, it would appear that such a carefully coded RC4 implementation cannot be attacked using SPA using the *single* available sample.

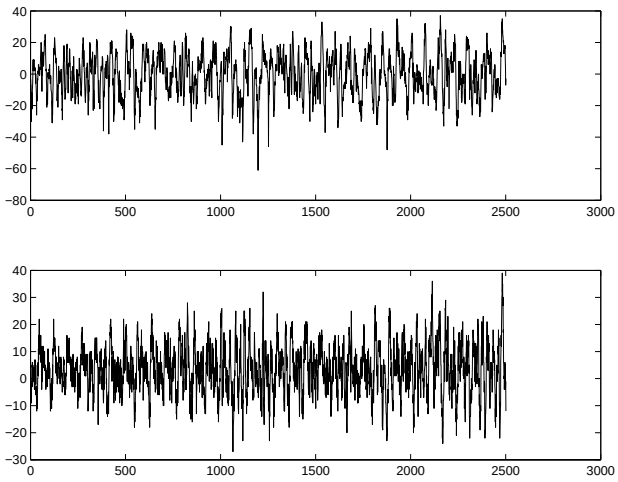

Figure 7: Differences of power samples: upper figure shows sample difference for same key, lower shows difference for two different keys.

RC4 is, however, an ideal candidate for template attacks. It is evident from inspecting the code snippet above that the key byte used in each iteration causes substantial contamination. The loading of the key byte, the computation of index `i2` and the use of `i2` in swapping the bytes of the state table T all contaminate the side–channel at different cycles. Averaging over a large number of samples makes the extent of this contamination easily visible by highlighting significant and widespread differences for two different

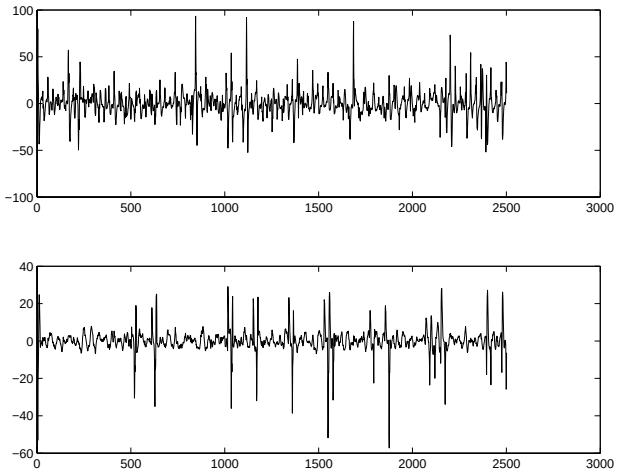

Figure 8: Differences of averages of 5 power samples: Upper figure is for the same key and the lower is for two different keys.

values of the key byte. Further, the use of `i2` and the state in subsequent iterations, and the fact that RC4 is a well-designed stream cipher, quickly propagates small key differences to cause diffusion. Thus, one expects that templates corresponding to different choices of key bytes are very different and can be used to efficiently and effectively classify a single sample.

We have obtained very good results in using template attacks to break this implementation of RC4 [5]. Inspecting the averaged RC4 side–channel samples using several different keys, we identified 42 points in the side–channel sample in each iteration where significant differences appeared. These are the places where the key has good contamination and thus these points are well suited to classify keys. Our first attempt used statistical models that treated these 42 points independently, i.e, we only looked at means and standard deviations of the samples at each of the points. Although encouraging results were obtained for distinguishing between pairs of key bytes which were very different (error close to 0%), there were unacceptably high classification errors (as much as 35%) for pairs of keys with only a few bit differences. Thus this approach was deemed unsuitable for an extend-and-prune attack due to high errors which would result in large number of potential candidates being retained.

Next, we used the Multivariate Gaussian Noise model. For our experiment, we used 10 choices for the first key byte. These were carefully chosen to be very close and therefore yielded poor results with univariate statistics. For each of the 10 values of the key byte, we took 2000 samples of the side–channel. We used the same 42 points of interest as in the univariate experiment. The templates consisted of the means and the noise covariance at these points. We used the templates and the maximum likelihood estimator to classify each of these 20000 samples. Our experiments showed that the multivariate Gaussian noise model was able to correctly identify the right key byte out of the ten possibilities with an average classification success probability of 99.3% and worst case success probability of 98.1% (sample chosen randomly among 2000 samples with same key byte). Since the 10 key bytes were carefully chosen to reflect the worst case, these results can be extrapolated to the case of 256 different values of the key byte. Even if we pessimistically assume for each key byte there are $50-60$ bytes which are *close*, we would get an average classification error of $5-6\%$ while classifying all possible values of that byte.

With the multivariate approach, we can extract small sized keys, even by using the drastic pruning strategy of retaining only one possibility for the key byte at each iteration. If the key is small, we need to run this process only for a few iterations and the overall error, bounded by the sum of the errors in each iteration, will still be small. For example, we can do better than 50% (total error = $8\times 6\%$) for about 8 bytes of key material.

With a little more effort, much better results can be obtained. We could keep more candidates at the end of each pruning stage and significantly reduce the proba-

bility of error. For example, with a slight modification of the maximum likelihood method, we can keep at most 1.3 hypotheses (on an average) for the key byte and have a 98.67% guarantee that the correct byte is retained. Using this approach independently for each iteration, for an n-byte key, we can reduce the number of possibilities down to $(1.3)^n$ while retaining the correct key with probability at least $(100 - 1.33n)\%$. Thus, we are able to substantially reduce the $8n$ bits of entropy in the key to about $0.38n$. The results of an actual template attack will be better than these estimates since the templates will built for longer and longer prefixes of the computation and not independently for each iteration. Due to the diffusion property of the RC4 key initialization algorithm, candidates having incorrect values for the initial key bytes are likely to be eliminated at later stages of the process. Thus the final number of candidates will be substantially lower than $(1.3)^n$.

References

[1] D. Agrawal, B. Archambeault, J. R. Rao and P. Rohatgi. The EM Side Channel(s). Proceedings of CHES 2002, Springer, Lecture Notes in Computer Science 2523, B. Kaliski, C. K. Koc, C. Paar (Eds.), pages 29–45.

[2] D. Agrawal, B. Archambeault, J. R. Rao and P. Rohatgi. The EM Side Channel(s): Attacks and Assessment Methodologies. See http://www.research.ibm.com/intsec/emf-paper.ps.

[3] A. V. Borovik and C. D. Walter. A Side Channel Attack on Montgomery Multiplication, private technical report, Datacard platform seven, July '99.

[4] S. Chari, C. S. Jutla, J. R. Rao and P. Rohatgi. Towards Sound Countermeasures to Counteract Power–Analysis Attacks. Proceedings of CRYPTO '99, Springer, Lecture Notes in Computer Science 1666, M.J. Wiener (Ed.), pages 398–412.

[5] S. Chari, J. R. Rao and P. Rohatgi. Template Attacks. Proceedings of CHES 2002, Springer, Lecture Notes in Computer Science 2523, B. Kaliski, C. K. Koc, C. Paar (Eds.), pages 13–28.

[6] J.-F. Dhem, F. Koeune, P.-A. Lerox, P. Mestr'e, J.-J. Quisquater and J.-L. Willems. A Practical Implementation of the Timing Attack. Proceedings of CARDIS '98, Springer, Lecture Notes in Computer Science 1820, J.-J Quisquater, B. Schneier (Eds.), pages 167–182.

[7] Dynamic Sciences International Inc. See http://www.dynamic-sciences.com/r1550.html.

[8] P. N. Fahn and P. J, Pearson. IPA: A New Class of Power Attacks. Proceedings of CHES '99, Springer, Lecture Notes in Computer Science 1717, C. K. Koc, C. Paar (Eds.), pages 173–186.

[9] L. Goubin and J. Patarin. DES and Differential Power Analysis (The "Duplication" Method). Proceedings of CHES '99, Proceedings of CHES '99, Springer, Lecture Notes in Computer Science 1717, C. K. Koc, C. Paar (Eds.), pages 158–172.

[10] NSA TEMPEST Documents. See http://cryptome.org/nsa-tempest.htm.

[11] K. Gandolfi, C. Mourtel and F. Olivier. Electromagnetic Attacks: Concrete Results. Proceedings of CHES 2001, Springer, Lecture Notes in Computer Science 2162, C. K. Koc, D. Naccache, C. Paar (Eds.), pages 251–261.

[12] P. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS and Other Systems. Proceedings of CRYPTO '96, Springer, Lecture Notes in Computer Science 1109, N. Koblitz (Ed.), pages 104–113.

[13] P. Kocher, J. Jaffe and B. Jun. Differential Power Analysis: Leaking Secrets. Proceedings of CRYPTO '99, Springer, Lecture Notes in Computer Science 1666, M.J. Wiener (Ed.), pp 388–397.

[14] T.S. Messerges, E.A. Dabbish, and R.H. Sloan. Power Analysis Attacks of Modular Exponentiation in Smart Cards. Proceedings of CHES '99, Springer, Lecture Notes in Computer Science 1717, C. K. Koc, C. Paar (Eds.), pages 144–157.

[15] P. L. Montgomery. Modular multiplication without trial division, Mathematics of Computation, 44 (1985), no 170, pages 519–521.

[16] J.-J. Quisquater and D. Samyde. ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smart Cards. Proceedings of E-smart 2001, Springer, Lecture Notes in Computer Science 2140, I. Attali, T. P. Jensen (Eds.), pages200-210.

[17] W. Schindler, A Timing Attack against RSA with Chinese Remainder Theorem. Proceedings of CHES 2000, Springer, Lecture Notes in Computer Science 1965, C. K, Koc, C. Paar (Eds.), pages 109–124.

[18] The complete unofficial TEMPEST web page. Available at http://www.eskimo.com/~joelm/tempest.html.

[19] H. L. Van Trees. Detection, Estimation, and Modulation Theory, Part I. John Wiley & Sons. New York. 1968.

[20] C. D. Walter and S. Thompson. Distinguishing Exponent Digits by Observing Modular Subtractions. Proceedings of CT-RSA 2001, Springer, Lecture Notes in Computer Science 2020, D. Naccache (Ed.), pages 192–207.

ABOUT RSA LABORATORIES

An academic environment within a commercial organization, RSA Laboratories is the research center of RSA Security Inc., the company founded by the inventors of the RSA public-key cryptosystem. Through its research program, standards development, and educational activities, RSA Laboratories provides state-of-the-art expertise in cryptography and security technology for the benefit of RSA Security and its customers.

Please see www.rsasecurity.com/rsalabs for more information.

NEWSLETTER AVAILABILITY AND CONTACT INFORMATION

CryptoBytes is a free publication and all issues, both current and previous, are available at www.rsasecurity.com/rsalabs/cryptobytes. While print copies may occasionally be distributed, publication is primarily electronic.

For more information, please contact:

cryptobytes-editor@rsasecurity.com.

RSA SECURITY®

RSA Security Inc.
www.rsasecurity.com

RSA Security Ireland Limited
www.rsasecurity.ie