THE ONASSIS FOUNDATION SCIENCE LECTURE SERIES

The 2010 Lectures in Computer Science: Network and Information Security

Summary report

Student:

Bojan Furlan

Institution:

School of Electrical Engineering,

University of Belgrade

Random Graphs in Cryptography

Prof. Adi Shamir, the Weizmann Institute of Science, IL

Cryptographic research can be classified broadly as *Information-theoretic* and *Complexity-theoretic*. Information-theoretic approach assumes that a.) Primitives are perfect and b.) Opponent is all powerful (in sense of computing resources), where it tries to bound: a.) Statistical properties and b.) Derived information. Examples are OTP and secret sharing. On the other hand, Complexity-theoretic approach assumes that a.) Primitives are imperfect and b.) Opponent is bounded, where the aim is to limit a.) Runtime of attack and b.) Required memory. Examples are AES and RSA key exchange. But there is a third type, which combines these two with assumption that a.) Primitives are perfect and b.) Opponent is bounded with the aim to bound a.) Runtime of attack and b.) Required memory. Example of this type of research is finding collisions or inverting edges in random graphs.

Many cryptographic processes can be modeled by random graphs and efficient algorithms for the analysis of random graphs can be used to speed up many general-purpose cryptanalytic attacks (including the analysis of the cycle structure of stream ciphers, the discovery of collisions in hash functions, the application of time/memory tradeoffs to block ciphers, etc.). The interface between cryptography and random graphs is a fascinating subject which had been studied for many years, but recently special emphasis is made on cycle detection algorithms and inversion algorithms based on graph covering techniques.

The notion of random functions (oracles) over the finite domain {0,1,2,...,N-1} has following properties: *f(x)* is truly random when applied to fresh inputs and is consistent when applied to previously used inputs. Therefore, we can associate random graph with f(x) where x->*f(x).* Interesting algorithmic problems in breaking the security of hash functions are finding *simple collisions* (assuming that we can only choose random points and move forward along graph edges) or finding *multicollisions* (e.g. useful in breaking concatenated hash functions).

One of algorithms for finding simple collisions is Floyd's two finger algorithm. The algorithm is based on keeping two pointers that are iterating in a random graph. One pointer is running at normal speed, and the other one is running at double speed, until they collide. Then for finding the entry point into the cycle, we first need to move one of the pointers back to the beginning and then iterate both pointers at equal speed. This algorithm is not so efficient because when the cycle is short, the fast pointer can traverse it many times without noticing.

Other elegant solution was published by Gabriel Nivasch in 2004. Advantages are that it uses a single finger and negligible amount of memory; it stops almost immediately after recycling; it is efficient for all possible lengths of cycle and tail and ideal for fast hardware implementations. The basic idea of the algorithm is to maintain a stack of values, which is initially empty. Each new value is inserted at the top of the stack, with forcing them to be in monotonically increasing order. Iteration is stopped when two identical values appear at the top of the stack. Because the smallest value on the cycle cannot be eliminated by any later value, the second occurrence will eliminate all higher values separating them on the stack. Therefore, the algorithm always stops during the second cycle, regardless of the length of the cycle or its tail and the maximal size of the stack is expected to be only of logarithmic complexity of the path length, requiring negligible memory. Improvement is done by partitioning values into k types, and using different stacks for each type. Iteration stops when repetition is found in some of the stacks.

Unlike Floyd's algorithm, the Nivasch algorithm provides excellent approximations for the length of the tail and cycle as soon as a repeated value is found, without extra work.

How to Exploit a Small Cryptographic Leakage

Prof. Adi Shamir, the Weizmann Institute of Science, IL

Side channel attacks are based on information gained from the physical implementation of a cryptosystem. This kind of attack is considered more powerful than the classical ones and is of great concern today. In many cases it is the only practical way to break well designed cryptosystems. However, since each side channel attack has unique characteristics (needs different physical and mathematical approach), the goal is to develop a new unifying framework which could be applied to all sorts of devices, without requiring detailed knowledge of the physical and logical implementation of the cryptosystem. Some examples of possible scenarios are: probing any wire in a chip; measuring the total power consumption; or using a tiny antenna to measure the RF field near the surface of the chip.

Block ciphers are typically iterated, applying the same operations in each round to different values, so any type of physical leakage is likely to repeat itself in each round, and all these values will be available to the cryptanalyst. The Cube Attack, published by Dinur & Shamir in 2008, is applicable to a wide variety of symmetric-key algorithms. One example is an attack to the Advanced Encryption Standard 128 (AES-128), since the AES-128 block cipher has a fixed block size of 128 bits and a key size of 128 bits. The attack only uses the plaintext and a single state bit leaked from the end of the second round in multiple encryptions, as it depends on all the 128 key bits. In more detail, attack is based on the fact that any cryptographic scheme can be described by multivariate, huge polynomials that contain both secret and public bits. These polynomials can be faced as black box polynomials. According to this approach, the problem of cryptanalysis is converted to solving a system of such black box polynomials. As a result, the cube attack is summing an output bit value for all possible values of a subset of public input bits, chosen such that the resulting sum is a linear combination of secret bits. Repeated application of this technique eliminates all nonlinear products and gives a set of linear relations between secret bits that can be solved to discover them. The shorter formula then leaks bits of the cryptographic key.

This attack is completely practical, requiring about 2^{35} for complete key recovery and the mathematical part of the attack was simulated successfully on a single PC in a few minutes. It can be applied directly to arbitrary black box polynomials and to unknown (or partially known) cryptographic schemes given as black boxes.

Identity, Security, and Privacy

Prof. Steven Bellovin, Columbia University, USA

Today, fundamental issues regarding identity, security and privacy are strategies used to assure protection of systems, networks and data. Generally speaking, these strategies can be summarized in 3 classes: by building better walls (on the level of operating systems, firewalls or applications), by encryption and by authentication. Some of the reasons why authentication is needed are to restrict access to some resources or to encrypt data to the right party. But it also imposes accountability and anonymity threads. One example is governments that just want to restrict freedom of speech and access – and even in democratic societies, there are abuses of anonymity. Anonymity can be a powerful force for good; It permits “whistleblowers” to disclose government or corporate wrong-doing, thus it should be strongly protected by law!

Some of the current cybersecurity threats are: Hackers – these days they are mostly motivated by profit; Industrial espionage – quite possibly sponsored by governments; foreign government espionage and

cyberwarfare (if there is such a thing)? Consequently, question that comes up is: Will strong authentication help against any of these?

Today, hackers usually don't use their own machines for most of their work; instead, they create botnets – armies of "bots"; so they are demonstrably capable of running arbitrary code on many computers belonging to many innocent people. Moreover, they steal all sorts of authentication credentials, so why should a new authentication scheme be stronger? On the other hand, governments effectively control all CAs within their jurisdiction. If a government wishes to issue fake credentials to spies or to industrial spies benefitting its own country's businesses – it will do so - there are many reports of fake passports issued by intelligence agencies. Therefore, no government will trust credentials issued by another government. Also, trusted hardware like smart card or TPM chip can't talk directly to the outside, instead they speak via the operating system. But operating systems are very vulnerable to attacks – which mean that trusted hardware can be controlled by the attackers. For all of these reasons, authentication doesn't solve adequately the cybersecurity problem, but it still provides some kind of protection.

Some real word issues that arise regarding authentication are how to authenticate people, how to deal with lost credentials or compromised ones. Also, privacy issues are: when the same pseudonymous identity is used in different contexts, a profile of the user can be built up; one link to a real person can tie a real person's activities to that person and such tracking can be and is being done by many parties. Therefore, to protect privacy, use of identity-linked credentials should be avoided. Rather, use of authorization credentials is preferred - the bearer has certain rights, regardless of identity and each use has its own credential. One way in this direction is the use of Unlinkable Credentials. Each user has a master key pair. The master private key can be used to generate subcredentials – a key pair that is verifiably derived from a given CA-issued certificate. Subcredentials cannot be linked to each other or to the master credential. Knowledge of a private subkey reveals the master private key. Advantages are strong authentication, pseudonymity – as many (or as few) pseudonyms as needed, privacy, no accountability and no revocability in case of private key compromise.

As a final point, some currently related policy and computer science questions are: What is the right tradeoff between societal interests in accountability and privacy? What is the proper cost – temporal, financial, and procedural – for revoking anonymity? If there is a revocability feature, how is it protected? How do we prevent leakage via lower-level (i.e., network layer) or higher-level (login name, writing style, interests) channels?

The Cybersecurity Challenge

Prof. Steven Bellovin, Columbia University, USA

The cyber-security challenge: we currently can't argue that cyberspace is more secure than it was 35 years ago, and even worst, we can't certainly say that this will get better soon. Most security problems are due to buggy code. Today, the code is better than 35 years ago – but the systems are far more complex, and the rate of complexity – and hence bugginess – has increased faster than the code quality. We're out of ideas! - There haven't been any fundamentally new defensive ideas for a long time. The basic mechanism is the wall – a barrier between good and bad programs, individuals, systems, etc. Walls are far from perfect, but the hard part is not the walls, it is the gates – the way we permit things to pass through the wall in a controlled fashion.

Some of the reasons why firewalls fail to assure protection are because of too many complex things allowed to pass through (Javascripts, PDFs, etc.); in addition, too many machines – laptops,

smartphones, etc. – live both inside and outside the firewall, introducing vulnerability. Also, leaks in operating systems walls are caused by lightly granted privileges (in other words, it is a form of gate) and the boundaries between trusted and untrusted components have been blurred. Finally, there are many applications (mailers, browsers, PDF viewers or word processors) that are really like operating systems (allow untrusted input, programmability or resource management). Those components are not part of the traditional OS, but failures of their protection schemes can result in user account penetration. As a result, walls will fail in unpredictable ways; intrusion detection systems are imperfect and the increased amount of connectivity, through and around firewalls, have rendered them essentially. Accordingly, there is a need for a new approach!

The current thread model can be described by profit-driven hacking, plenty of resources being devoted for attacks (bots), easy to exploit applications (developed on a very tight budget and schedule), rapid introduction of new devices and applications (iPhone, Facebook, Twitter) - and as so, new vulnerabilities. Asymptotically, computers, bandwidth and disk space are free, so there is no great value to protect them. But people, physical world or data (especially in the aggregate) are valuable and expensive. By a personal vision of professor Bellovin, future research that could lead to a solution, can be divided into four themes: resilience, usability, large scale systems and modes of thoughts. A resilient system can be defined as one that protects most of its data most of its time – low enough rate of data protection failure. On the other hand, usability is an important factor regarding that many of today's security systems are just too difficult to use. Also, having in mind that today's systems are consisting of many interconnected systems - where each is a potential point of vulnerability - instead of defense in depth, there is a weakness in depth. Therefore, we need ways to understand the properties of systems as whole and ways to manage the security settings – including configuration and patch level – of large-scale systems, without very much expensive, buggy human intervention.

Finally, we don't know how to think about new threats or new services and as a result we approach the questions in an ad-hoc fashion and try to reason with analogy. But the threat will change, depending on the application. How could this be anticipated? The usual approach is extremist: either there are no problems, or all new services are banned. Generally speaking, both are incorrect – but what should replace them? Is it possible to have a useful formalism that can describe things that haven't been invented yet? We have to try something new!

Diagnosing the genuineness of events in runtime monitoring of security and dependability **&** Runtime prediction of security and dependability threats: The EVEREST framework

Prof. George Spanoudakis, City University London, UK

The main objective of the SERENITY approach is dynamic assembly, deployment, selection and (re-) configuration of components that can realize Security and Dependability (S&D) solutions in applications driven by S&D patterns. The motivation behind the SERENITY implementation was a large variety of applications that have continually changing S&D requirements or operational environments and contents, and because of the need to interact with dynamically assembled distributed components as well.

S&D patterns provide an abstract specification of solutions that can be deployed in a system to provide S&D properties and link this specification to alternative concrete implementations. An example of

SERENITY framework use is Location based Access Control System. This system provides access to enterprise resources (e.g. printers or Internet access) from mobile user devices (PDAs, laptops) where it implements Device Location Pattern that calculates the position of a device with some accuracy measure. In this kind of applications, runtime monitoring of S&D solutions is required in order to check preconditions and invariants required for the correct operation of the solutions. Also, it verifies dynamically that an S&D solution operates according to its specification in all circumstances (static verification and testing cannot provide a full guarantee for this); predicts possible violations of conditions and takes (if possible) preemptive actions. The conditions that must be examined in order to monitor the runtime are: the availability of the location server, the liveness of signal daemons in mobile devices and the accuracy of location information.

In the SERENITY monitoring infrastructure checks are not performed by an application itself, but by the external entity - SERENITY Runtime Framework (SRF). Therefore, this framework requires monitoring specifications, so it is more flexible when operational environments and S&D solutions change, and can be applied to external collaborators, but it is less efficient than application based testing.

Briefly, EVEnt REaSoning Toolkit (EVEREST) captures events through event captors associated with systems and their components; checks whether captured events (and events deduced from them) satisfy specific S&D properties expressed as monitoring rules (core monitor); assesses event genuineness by attempting to derive explanations of captured events (diagnosis tool) and predicts potential violations of monitoring rules based on historical data (threat detection tool – TDT). In order to have satisfying monitoring conditions, it is obligatory to specify some monitoring rules about events - occurrences that happen within a system of instantaneous duration, e.g. receipt of component messages, execution of internal or system operations; as well as fluents - conditions about the state of a system, established by assumptions (deductive reasoning). Rules and assumptions are specified in Event Calculus and some examples are Happens, Initiates, Terminates, HoldsAt, etc.

Furthermore, the SERENITY approach provides some other advanced capabilities about diagnosis and threat detection. The diagnostic capabilities refer to the genuineness of an event, which depends on the ability to find a valid explanation, either by finding a logical combination of other events and states of the system which would have the event as a consequence, or it has as consequences other events which have also been observed and are genuine. Possible event explanations are generated by abductive reasoning using the monitoring specifications of the active patterns of the system that is being monitored. Event genuineness is assessed by beliefs computed according the Dempster-Shafer theory of evidence.

Threat detection is dedicated to detection of potential violations of S&D monitoring rules, by using DS beliefs to measure the likelihood of events genuineness and the likelihood of conditional event occurrence, negating the rule to get the exact pattern of events that violates it and constructing a belief network indicating how beliefs in the violation of the rule can be updated as partial evidence about events in the pattern emerges. Finally, reactions are realized by actions taken at runtime by the SERENITY Runtime Framework following the receipt of monitoring results from EVEREST. Each of the actions is executed only if the condition associated with it is also satisfied and the order of action execution is exactly the same as they appear in the rule specification. There is a set of predefined action types that are supported and complex conditions may be associated with actions as well.

In conclusion, SERENITY provides an infrastructure for selecting and deploying S&D solutions at runtime based on S&D patterns. It also provides a monitoring framework for runtime checks of conditions that are specified as monitoring rules in Event Calculus and are related to the correct operation of S&D

patterns. Diagnosis and threat detection capabilities (i.e., detection of potential violations of monitoring rules) are provided and accompanied actions are taken if some of those rules are violated. Further research is needed in order to extend predictive capabilities of EVEREST to support forecasting of violations of aggregate properties. In addition, support for protocols for reliable messaging (WS-ReliableMessaging) and message authentication (WS-Security) is needed, as well as support for evolution of S&D solutions both at the pattern and the implementation level.

Introduction to Provable Security for Symmetric Key Cryptography & Using Symmetric Key Cryptography to Build Secure Channels

Prof. Kenny Paterson, Royal Holloway University of London, UK

It is very important to guarantee the confidentiality and integrity of data travelling over untrusted networks (remote access, E-commerce or secure file transfers), hence there are many challenges in the design, implementation and security analysis of secure channel protocols. Concept of secure channel represents a channel that offers data origin authentication, integrity, confidentiality and anti-relay, but usually it is not offering non-repudiation and any security services once data has been received. Initialization is mainly carried in 3 steps: 1) an authenticated key establishment protocol - one or both parties are authenticated and a fresh, shared secret is established using asymmetric public key, or symmetric cryptography, or a combination of the two; 2) a key derivation phase - MAC and symmetric encryption keys are derived from the shared secret established during protocol; and finally 3) further traffic is protected using derived keys. MAC gives data integrity mechanism and data origin authentication, and confidentiality is provided by the encryption. Also, symmetric cryptography is further used for speed.

Security models for symmetric encryption are well established, especially for IND-CPA (Indistinguishable – Chosen Plaintext Attack) security. Informally, IND-CPA can be described as a computational version of perfect security, where ciphertext is leaking nothing about the plaintext. The stronger notion of this model requires that the adversary can also recover plaintext. IND-CPA security is easy to achieve by using suitable mode of operation of block cipher and is formalized as a security game between the adversary and a challenger. An example in order to achieve stronger security is Message Authentication Codes (MACs) algorithm - a symmetric analogue of a digital signature – that can be used to provide authenticity/integrity for messages. Its key security requirement is unforgeability, where an adversary cannot create the correct MAC tag for another message by having seen MAC tags for many other chosen messages.

IPsec provides encryption at the IP layer and it is cryptographic protection of IP packets and their payloads. A protocol that ensures this kind of encryption is the Encapsulating Security Protocol (ESP), which uses symmetric encryption and MACs. There are a few versions of ESP and every version provides advanced properties. One of attacks to IPSec is Linux ESP attack (Paterson and Yau) where the main idea is to exploit bit flipping weakness of CBC mode encryption, by capturing packets from network and modifying headers of inner packets, so that the error messages are produced when processed by IP and sent outside of IPsec tunnel (Error messages are carried by ICMP and reveal partial plaintext data). The modified inner packet is recovered upon decryption, and forwarded to the host indicated in destination address field. This host generates an ICMP “protocol unreachable” message in response to the (modified) protocol field in header and finally attacker intercepts ICMP message to get plaintext bytes. The characteristics of Linux Attack are that it recovers complete contents of IPsec-protected datagrams,

it is efficient and do not require special operating conditions. Another point of view was attack proposed by Degabriele and Paterson, based on extension of padding oracle attacks on CBC mode combined with previous techniques. In this case, attack should work if and only if implementer has followed all the check policies. Resulting attack is less efficient than Linux attack, but still recovers all plaintext.

Today, the theoretical cryptography community is well aware of the need to carefully combine integrity protection with encryption in order to achieve IND-CCA security and to prevent active attacks against confidentiality. It is also well-known amongst IPsec experts that encryption-only configurations should be avoided (Clear warnings against their use in the RFCs, Bellovin attack). But developers rarely pass RFC warnings to end users or don't properly implement RFCs, or end users just probably don't read RFCs and technical papers, so the danger still exists. Sometimes attacks that work in practice wouldn't work if the RFCs had been followed. On the other hand, the attacks that work on paper often don't work in practice against the RFCs. But there is a range of attacks against encryption-only ESP, so it is dangerously weak in a practical sense. This does not contradict theory: CBC mode is trivially insecure against a CCA attacker. But a fully practical attack seems necessary to convince practitioners for the need of CCA security. Even then, standards may not be changed, for many other reasons. In fact, little or no security is gained from the provision of upper layer integrity protection. The attacks arise from the interactions between cryptographic components and the "ambient" system in which they operate, so they illustrate the dangers of viewing cryptography in isolation only.

SSL (Secure Sockets Layer)/TLS (Transport Layer Security) is widely deployed protocol in Web browsers and servers to support "secure e-commerce" over HTTP. It runs over TCP, providing reliable, end-to-end transport. SSL/TLS Record Protocol provides data origin authentication, integrity and anti-replay using MACs and sequencing; and confidentiality using symmetric algorithm. The SSL/TLS Record Protocol is not generically secure; it is actually a MAC-then-pad-then-encrypt construction. Padding in SSL/TLS has a particular format by adding a sequence of bytes after the MAC in order to complete the last plaintext block. If t bytes are needed, then it adds t copies of the byte representation of t. This suggests that implementations should check the format of padding and terminate the connection if the padding format is incorrect. In order to defend against the attack, implementations MUST ensure that record processing time is essentially the same whether or not the padding is correct. In general, the best way to do this is to compute the MAC even if the padding is incorrect, and only then reject the packet. Padding oracles may seem esoteric, but can be difficult to avoid them in practice, e.g. in IPsec, an RFC-compliant implementation will check an extended padding format and silently drop the packet if check fails, and forward the packet if it passes and in SSL/TLS, an RFC-compliant implementation will issue an error message if the padding check fails; therefore it is up to the implementation to ensure that this does not introduce a timing side-channel.

In conclusion, there is a complex interplay between theory, specification and implementation in a domain of security. Cryptography is usually only a component in a larger system or protocol. So, implementing cryptography in real systems is fraught with dangers. The focus was almost exclusively on error based side channels and format-based attacks, but many other types of implementation-based attack are known.

The cryptographic hash function crisis and the SHA-3 competition

Prof. Bart Preneel, Katholieke Universiteit Leuven, BE

Cryptography is only a small subset of security, but an important one. Most systems are usually broken elsewhere, e.g. exploiting incorrect requirements or specifications, implementation errors, social engineering, etc. One of the principles that the cryptology is based on is "crypto box" that includes block ciphers which operate on fixed length blocks of plaintext and output a corresponding ciphertext. The transformation inside the cipher is controlled by the secret key that is used as a second input parameter. Some examples of symmetric-key encryption standards that use block ciphers are AES(128-192-256), RC6, DES(56), 3-DES(112-168), IDEA(128).

Beside block ciphers, another important concept is Message Authentication Code (MAC) algorithm that accepts as input a secret key and a message to be authenticated, and outputs a MAC that is a plaintext (signature). The MAC value protects both message's data integrity as well as its authenticity, by allowing verifiers (who also possess the secret key) to detect any changes to the message content. Example schemes are CBC-MAC, HMAC. Also, there is another public key cryptology approach, based on digital signature, where verifiers use a public key that is different from the private key used for signing. Whole process is much slower, however it does not require shared secret key. Example schemes are RSA, DSA, ECDSA.

Underneath the whole process of signing lays an important theory of cryptographic hash functions. An input to a hash function is a string of an arbitrary length and an output is a short fixed bit length string. The main property of cryptographic hashing is that it should be very hard to find an input for a given value (a preimage) or to find two colliding inputs (a collision). Two classes of cryptographic hash functions are MAC and MDC (OWHF, UOWHF, CRHF).

Preimage resistance refers to the difficulty of finding a preimage x that produces a specific hash h(x). Second preimage resistance refers to finding a preimage y different of a specific unknown x, satisfying that h(y) equals to known h(x). Finally collision resistance refers to finding two different $x \neq x'$ satisfying that $h(x) = h(x')$. Preimage resistance doesn't imply 2^{nd} preimage resistance and opposite, but collision resistance implies 2^{nd} preimage resistance.

One of 2^{nd} preimage multiple targets attacks exploits the fact that if one can attack 2^t simultaneous targets, the effort to find a single preimage is 2^{n-t}. The answer on this brute force attack is the randomization of hash function with a parameter S (salt, key, spice). On the other hand, the brute force collision search based on "the birthday paradox" uses the fact that for a given set of S elements and chosen r elements at random (with replacements), where $r \ll S$, the probability p that there are at least 2 equal elements (a collision) is then $p \cong 1 - \exp(-r(r-1)/2S)$. There are also many other brute force collision search attacks, thus collision resistance is hard to achieve in theory and practice. Fortunately, collision resistance is not always necessary and there are other properties that are needed in practice, such as pseudo-randomness if keyed (with secret key), near-collision resistance, etc.

For hashing very large strings, message can be split into fixed length blocks and then hash function is made by an iterated structure, repeating a compression function over each block. Nevertheless, this kind of iteration can degrade security and solution to some extent is using Merkle-Damgard (MD) strengthening (using fix IV, unambiguous padding and inserting length at the end). Output transformation (use of extra function g at the end of iteration) is a solution for the problem of length extension. Also, concatenation of two or more functions intuitively could strengthen hash function, but

in reality it does not. Actually, MD iteration should be improved using “salt” + output transformation + counter + wide pipe.

In conclusion, most of the industry standards like SHA-1 and MD5 should not be used anymore, as they are compromised. SHA-256, SHA-512 and RIPEMD-160 are more promising, but an open SHA-3 competition should yield new standard as result of 2009-2012 research insights. The ultimate goal that still stands is an efficient hash functions with security reduction.

Identity management and privacy

Prof. Bart Preneel, Katholieke Universiteit Leuven, BE

Information storage and transmission grows exponentially, followed by computing power and the number of information processing devices (the Internet of things, ubiquitous computing, pervasive computing and ambient intelligence). Therefore, the need of identification represents crucial part of the system. The uniqueness can be identified at physical and electronic level (radio fingerprinting, fibers in paper or magnetic behavior of certain materials) or on a level of human biometry (fingerprint, iris, DNA). Furthermore, there are some deliberate attempts for identifying electronic devices like MAC addresses or IMEI. Therefore, people can easily be located by a cell phone, laptop or credit card. This enables new possibilities for location based services, such as traffic monitoring and emergency services, location finders (for nearest restaurant, gas station, etc.) or social applications (Geotagged Twitter, Google Latitude). With these new technologies the problem of privacy arises - to be seen at certain locations, such as AIDS clinic, business competitor, or political headquarters can be compromising. Moreover, some critical information can be easily inferred from this kind of trace (desk in a building, home location or future locations).

The privacy debate raised many questions, some believed that care about privacy is related to hiding something, others advocated the need for a tradeoff between privacy and security and also, there was an opinion that people don’t care much about privacy. On the other hand, privacy is not only about hiding bad things and the necessity for more surveillance is a strong argument but not effective, the risk of abuse and subversion is prevailing. Finally, people want to control personal information due to impression management /self-presentation or personal safety, so they care about privacy.

Privacy is a security property for individuals, companies, governments or infrastructure. It is an abstract and subjective concept that can be described by confidentiality, control or practice. Recent definition of privacy by (US) National Strategy for Trusted Identities in Cyberspace includes the appropriate use of personal information under the circumstances and also, the right of an individual to control the collection, use, and disclosure of personal information. On a legal basis data should be protected and only collected for specific purpose, in a proportional way with the subject’s awareness and consent. Broadly, two privacy models can be distinguished: soft and hard. Soft privacy means that subject provides data to data controller, which is responsible for its protection. Data subject cannot verify how data is collected and processed because she has already lost control by providing it. On the other hand, hard privacy means that subject provides information as little as possible, reducing the need to “trust” other entities.

Identity Management (IDM) is the secure management of the identity life cycle and the exchange of identity information (e.g., identifiers, attributes and assertions) based on applicable policy of entities, such as: users/groups; organizations/federations/enterprise/service providers or objects (application process, content and data). Identity relates to a dynamic collection of all entity's attributes (1 entity: 1 identity) and partial identity is a specific subset of relevant attributes. Therefore, IDM has many dimensions so it is not sufficient to add an "identity layer" to the Internet.

First step (identity 1.0) in IDM roadmap is centralization, by integrating entity authentication, embracing multiple authoritative sources and making account names ephemeral with dynamic, not static rules. Step number 2 (identity 1.5) demands federation. Federated identity means that credential of an entity that links an entity's partial identity in one trust domain to an entity's partial identity in another trust domain. One model is Single Sign-On (SSO) where relying parties don't ask IDP to re-authenticate subject, which implies that user logins only once. An example of SSO with symmetric keys is Kerberos. This model (SSO) is more convenient and secure than multiple passwords, but there are some problems like privacy risks (central control of who accesses which services at which time). Main issues of identity 2.0 are the need of consistent view for the user: the identity selector and the move from enterprise centric to user-centric privacy (user in control).

In conclusion, it should not be forgotten that users, businesses and government are players in the same game, but with different goals. Privacy is not "opposed" to security, it is a security property. One of pessimistic predictions is that security for society will grow, but privacy of individual will erode.

How Cryptosystems Are Really Broken

Public Lecture by Prof. Adi Shamir, the Weizmann Institute of Science, IL

Purpose of cryptosystems is to send plaintext securely from one point to another. On one side of communication channel is an encryption device, which produces ciphertext from plaintext and on the other, a decryption device that recovers plaintext from received ciphertext. The mathematical "black box" model of cryptography assumes that cryptanalyst has an access to ciphertext and, occasionally in some scenario, has an access to plaintext (either on encryption or decryption side). Therefore, mathematical cryptanalysis, like breaking the German ENIGMA and Japanese PURPLE, had a major impact on the outcome of WW2. However, modern cryptosystems cannot be broken with such mathematical techniques. Today, there is a much better understanding of how to construct cryptosystems that can resist to the all known types of mathematical attacks. Also, use of faster microprocessors for braking codes can be prevented by using more complicated cryptosystems with longer keys.

So in theory, cryptanalysts should be out of work… But as an expression says: "Only in theory, theory and practice are the same", thus struggle for braking cryptosystems remains, but on different levels. One aspect is a key stealing technique, such as Espionage (e.g. The Walker family of spies); Trojan horses: Capturing passwords entered into PC's; Tampered cryptosystems: The Swiss company CRYPTO AG; etc. A new technique (published in 2008) called "cold boot" extracts disk encryption keys from a lost or stolen laptop that has important data, protected by a disk encryption program such as bitlocker. One of the assumptions is that the encryption scheme is the strong AES and the stolen laptop is in a sleep mode,

where resuming operation requires a long unknown password. The AES encryption key is kept in the volatile RAM inside the laptop, which is erased if the computer is turned off or when the battery runs out. The braking technique is based on observation that the data deteriorates over time in unpowered RAM, but at a rate that *depends on the temperature*. Therefore, data can be kept alive for many seconds in unpowered RAM by cooling it with a Quick-Freeze can. The procedure for recovering keys after cooling the RAM chips consist of: Rebooting the laptop via a small operating system located in a disk-on-key; Quickly dumping the memory contents into the disk-on-key; Analyzing the data to find a slightly corrupted AES key and, at the end, using the fact that the 128-bit key is expanded in memory into 10 related 128-bit subkeys, which forms an excellent error correcting code.

There is another "grey box" view of cryptography, which is using so called side channel attacks for key retrieval. Side channel attacks utilize information gained from the physical implementation of a cryptosystem, like acoustic leakage or power consumption variation by the hardware during computation. The power consumption can be easily recorded by using the USB connector, as it supplies both power and data to external devices. One attack on the RSA scheme (published in 2007) is exploiting such power traces. To decrypt ciphertexts or sign messages, the device computes $x^d \pmod n$ where d is the secret RSA key. Since d is very large, the exponentiation is typically done by a sequence of squaring (S) and multiplying (M). Therefore, key can be easily retrieved if it is possible to distinguish between S and M. In the past, those two operations were implemented by very different algorithms, which made it easy to distinguish them by just looking at the power consumption curve. Today, this is no longer true, and to make a distinction it seems that a large number of curves and sophisticated signal processing is needed. But there is an exceptionally simple way that is based on idea that comparing two curves can serve as an equality oracle. If we multiply a*b and c*d, then the two curves will look similar if a=c and b=d, and different otherwise. Therefore, the goal is to perform only two exponentiations and to compare the corresponding segments in the two power consumption curves.

Finally, new types of side channel attacks are found and published every few months. Recent discoveries of side channel attacks far outnumber those of classical cryptanalytic attacks and also are much more practically significant. Therefore, the issues of how to develop and implement new crypto applications and how to formally prove their security are still open.

Security & Privacy on the Mobile Internet - Part I **&** Part II

Dr. Udo Helmbrecht, Executive Director of ENISA, EU

According to the EU analysis of IT-Security, many of the risks are expected to increase or at least to remain the same in the near future. Also, their economic dimensions are grooving, so European Commission launched the EU 2020 strategy where several goals have been set in order to ensure IT safety. Some of these steps include the modernization of ENISA (European Network and Information Security Agency); the enhancement of the cooperation between CERTs (Computer Emergency Response Teams) on national & European level; Support EU-wide cyber security preparedness exercises and enhance prevention and combating cybercrime. Several technological areas with an impact on resilience were analyzed, showing that most IT-related risks are generated by the excessive automation and the vulnerabilities of the technology itself (e.g. due to diverse levels of resilience in interconnected networks and devices, increases in attack surface – down to firmware and up to applications, exponential increases in traffic, or regulatory disconnections between different geographical areas). Some examples of these technological areas are: cloud computing, future wireless networks, sensor networks and supply chain integrity.

In Cloud Computing, shared resources are provided and easily released on demand. The cloud model is composed of five characteristics (on-demand self service, broad network access, resource pooling, rapid elasticity, measured service), three service models (SaaS - cloud software as a service, PaaS - cloud platform as a service, IaaS - cloud infrastructure as a service), and four deployment models (private, community, public or hybrid cloud). Main advantages are that it promotes availability, offers a high degree of redundancy and allows economies of scale. But on the other hand, the cloud computing paradigm changes the threat landscape with respect to both service availability and the protection of sensitive data. Also, resiliency has become an important concern in Future Wireless Networks (MANETs, WMNs). These networks are usually protected by authentication, access control and cryptographic algorithms. However, their dynamic topology, peer-to-peer architecture, and shared-medium access set new challenges to the design of networking mechanisms that improve resiliency. Some of them are: protecting route discovery, reactive distance vector routing, proactive link-state routing, protecting resource reservations and design issues in error recovery mechanisms. Another point represents Sensor Networks that are widely installed worldwide in urban, suburban and rural locations and the Internet of Things is expected to convert them into first class devices, fully visible with an end-to-end connectivity. Since some of these sensors operate over critical infrastructure, they should be protected effectively, offering increased security, privacy and confidentiality. Finally, Supply Chain Integrity is an important topic in ICT and currently, is addressed separately in different industries. Managing supply chain integrity risks by taking into account several challenges, such as the absence of common business models, the chain complexity and the lack of broadly applicable tools and techniques to detect or defeat counterfeiting and tampering in systems, is still an open issue.

The term “mobile computing” is very generic and includes several devices like notebooks, hand-held computers (PDA), or smart phones. These devices have different characteristics and are used in different ways, but many of the key risks are the same. Some security issues relate to the following properties: wireless transmission is utilized by weak encryption mechanisms; user interfaces are more primitive due to device constraints; battery life is a concern for encryption; numerous personal data are held and finally, these devices are easier to be stolen. As a result, security in mobile devices is of great concern and some basic concepts include the GSM, IEEE 802.11 and Bluetooth technologies, as well as their encryption mechanisms and their weaknesses. Smart phones because of their capabilities, such as full Internet access, support for data storing and a wide range of applications, are designed to be used anywhere, so the security model must take this into account. However, standardization is extremely difficult, as well as implementation of cryptography due to the limitations on size and performance.

Finally, one of the solutions proposed for mobile computing problems is Elliptic Curve Cryptography. This approach is based on the arithmetic of points from an elliptic curve over a finite field. Its advantage over RSA algorithm is significantly shorter key length, consequently reducing power consumption and increasing performance.

Intrusion Detection Systems

Prof. Socrates Katsikas, University of Piraeus, GR

Intrusion Detection System (IDS) is a software or hardware product that monitors the events occurring in a computer system or network and analyses them for signs of intrusions. A concept of intrusion usually refers to attempts or attacks from the outside to break into or misuse the system. Therefore, Intrusion Detection is the art of detecting inappropriate, incorrect, or anomalous activity.
Generic Intrusion Detection Model consists of:

- Event generator, which provides information about system activities. Events are derived from system audit trails, network traffic, and application level systems.
- Rule set – The element that decides whether an intrusion has occurred. Events and state data are examined using rules, models, patterns and statistics in order to identify and flag intrusive behavior.
- Activity profile, which maintains the state of the system or network being monitored. Variables in the profile are updated as events appear from the monitored data sources.

Advantages of using IDS are that early detection may prevent, or at least minimize the extent of damage. Also, IDS existence can serve as deterrent to potential intruders, thus preventing them; and collection of information can be used to strengthen the prevention facilities or legally prosecute intruder as well.

IDS can be classified according to: the source from which data is collected for analysis; the detection mechanism by which the collected data is analyzed in order to detect potential intrusions; and finally, the response mechanisms that are triggered as a result of generated alerts. Source of collected data can be on application or host level, generally by analyzing the application log files or operating system audit trails respectively. On the other hand, data can be examined on network traffic level, which often consist of single purpose sensors that run in stealth mode. Some of advantages of Host-based over Network-based IDS approach is that it can collect vast amount of information for more reliable and precise detection and it is not affected by encryption or switched network, but on the other hand large number of monitored hosts can be harder to manage; it imposes overhead by using system resources and also attacks that target an entire network cannot be easily detected.

Detection methods can be broadly classified by following models: Misuse Detection - compares system activity to a predefined pattern of events that describes a known attack; Anomaly Detection that is based on the assumption that misuse or intrusive behavior deviates from historical norms; Specification-based Detection - uses rules that define the correct operation of a program/protocol and Hybrid approach that combines some of those models.

Damage caused by attacks is divided more or less equally between insiders and outsiders, but it is more difficult to detect insider attacks primarily because of their privileges. Insider is normally acting in accordance with commitments and duties, but occasionally mistakes may cause damage. Also when attacking, the attack is based on a well-prepared plan. Most prospective approach for this kind of threat is user profiling, which can uncover intentions. Therefore, an insider activity can be either normal (N), a mistake (M), act of a pre-attack phase (P) or an attack (A). These interactive situations can be modeled by Game Theory where intrusion prevention is based on an interaction between a user and the Intrusion Prevention System (IPS) that protects a Target System (TS).

The Economics of Privacy

Prof. Alessandro Acquisti, Carnegie Mellon University, USA

By some authors privacy can be defined as: Right to be left alone… (Warren and Brandeis 1890); an aspect of human dignity… (Bloustein 1964); or (Westin 1967) Informational privacy - "Privacy is the claim of individuals, groups or institutions to determine for themselves when, how, and to what extent information about them is communicated to others“. Many of those definitions are even contradictory, but ultimately, privacy relates to the negotiation/harmonization of private and public spheres: (Noam 1996) “Privacy is an interaction, in which the information rights of different parties collide. The issue is of

control over information flow by parties that have different preferences over 'information permeability'."

Since the privacy is about trade-offs: pros & cons of revealing and accessing personal information (for data subjects or data holders) and trade-offs are the realm of economics, hence, the privacy is an *economic* problem. Even when privacy issues may not have straightforward monetary interpretation and even when the entities involved may not be aware that they are, in fact, facing/accepting trade-offs there is a rational for studying these incentives and trade-offs that emerge from dynamics between public and private spheres. In addition, open issues are what conditions and what trade-offs maximize social and individual welfare and how to attain those conditions: Through the market? Through self-regulation? Through technology? Through legislation?

Privacy issues actually originate from two different markets, the market for personal information and the market for privacy, which are related, but not identical. Market for privacy mainly consists of companies that offer privacy enhancing technologies or companies that promise to keep their customers information protected and private, but also of consumers who adopt these technologies and/or strategies (e.g. identity theft insurance, anonymous browsing: "Freedom Network" or Tor). On the other hand, market for personal data involves companies that deal with customers' data, infomediaries, credit bureaus or companies that want to know more about consumers. Also, a big part of this market share consists of consumers who willingly or unknowingly reveal personal information. Some illustrations are: price for email addresses: ~few dollars for 100,000/1,000,000s; for computer as a slave in a botnet: few cents; for US SSNs: in the grey market: $5-$40 or in the black market: 50c to $20 (Perrig et al 2007 on the underground economy). Some valuations of privacy may alternatively be anchored by the expected price in the marketplace if data could be sold there; loss if data were compromised; cost of protecting data; profit for data holder from exploiting data.

Some theoretic models (Stigler, Posner and later Varian, Noam) conclude that sharing information between sellers reduces "distortions", so with "strategic" customers, firms will be better respecting customer's privacy. Therefore, economic modeling shows that free flow of certain information is good, and that market forces may tend towards optimal equilibrium. Consequently, question that arises is: Can all privacy problems be solved by the free market?

The study about how people really care for privacy shows dichotomy between privacy attitudes and privacy behavior (Spiekermann et al. 2001, Acquisti & Gross 2006's Facebook study). Some interesting remarks are that few people even read privacy policies, most give away personal data for small rewards and success of Facebook, Twitter, blogs and in general Web 2.0 suggests preference for information sharing over information protection. Accordingly, important issue was to analyze willingness to pay (WTP) versus willingness to accept to give data (WTA). This analysis (Tsai, Egelmann, Cranor, and Acquisti 2007) revealed that consumers are not always willing to trade-off privacy for monetary benefits!

Privacy, Behavioral Economics, and the Illusion of Control

Prof. Alessandro Acquisti, Carnegie Mellon University, USA

Contrasting (yet co-existing) human needs: need for privacy and need for publicity (even bad one), opens an issue of control that can be viewed as an economic signaling. Therefore, research evolved from the economics of privacy to the behavioral economics of privacy, where a rational model of privacy decision making mainly depends on WTA (benefits) and WTA (costs). Noticed obstacles that hamper (privacy) decision making are bounded rationality, incomplete information and some cognitive/behavioral biases.

Behavioral experimental economics has uncovered evidence for several systematic "deviations" from the theoretical rational behavior of the economic agent. Many of those deviations have applications to the privacy arena (as well as information security). Hence, the need arises for the application of behavioral experimental economics to understand how people make decisions about the security or privacy of their data and how cognitive and behavioral biases (negatively) affect those in order to inform policy and technology design. Methodological approach is based on testing hypotheses usually driven from BE and BDR theory, by randomly assigning subjects to different experimental conditions (e.g. different versions of a survey). Privacy concerns are a latent unobservable variable, so it is needed to observe manifest variables that are likely correlated (e.g., privacy-sensitive choices, willingness to share private information or likelihood of answering sensitive questions).

One of recent studies in behavioral economics of privacy deals with the impact on others of one's personal information. Aim is to answer two important questions: How does information related to past events and retrieved today get discounted? Does information about a person's past with negative valence receive more weight in impression formation than information with positive valence? Three survey-based randomized experiments were carried, manipulating valence of information provided to subjects and time to which that information referred: the dictator game, the wallet story and the company story. All subjects received the same baseline information about a person or a company and then subjects were asked to express a judgment on the person or company they just read about. The summary results showed opinion that bad is not just stronger than good; it is also discounted differently (Discounting the Past), with implications for future impact on revealed information.

Other research was dealing with the human propensity to reveal personal information. Tested hypothesis, named the Illusion of Control, states that users with more [less] control over disclosure and publication of personal information, but less [more] control over access and use of that information, disclose more [less] sensitive information, relative to status quo (Even though objective privacy costs derive from access to/use of information by others). Conjecture was that individuals may confound control over publication of private information with control over access/use of that information by others. Experiments were taken by reducing or increasing perceived control over publication of personal information (e.g. explicit vs. implicit control or certainty vs. probability of publication). Results showed that less perceived control over publication reduces revelation of private information and vice versa. This effect is stronger for more intrusive questions, which means that the publication of private information *per se* is not what disturbs people, but the fact that someone else will publish it for them.

As a conclusion, people's concerns for privacy (and security) depend, in part, on priming and framing. This does not necessarily mean that people don't care for privacy, or are "irrational," or make wrong decisions about privacy. Rather, it implies that reliance on "revealed preferences" argument for privacy may lead to sub-optimal outcomes if privacy valuations are inconsistent. People may make disclosure decisions that they stand to later regret; and those risks are greatly magnified in online information revelation. Therefore, these implications should steer debates about privacy regulation and policy-making.