When Cryptography is not the Answer (even when it is)

Orr Dunkelman

Computer Science Department
University of Haifa

20^{th} January, 2014

Cryptopia

- Since the 1970's our cryptography has matured:
  - Secret-key primitives (DES, AES, RC4, ...)
  - Public-key algorithms (DH, RSA, ECC, ...)
  - Zero Knowledge Proofs (a Turing-worthy idea)
  - Secure Multiparty computation
  - Traitor Tracing
  - Homomorphic encryption (the solution to all your problems)

Cryptodise

- Our understanding on how to use cryptography matured as well:
  - Provable security (reductions)
  - Standards
  - Protocols
  - Limitations* (e.g., privacy)

Cryptography is all around us these days

- SSL/TLS
- WPA2, WPA, WEP, ...
- IPsec/VPN
- Mobile communications — A5/1, A5/2, A5/3, A5/4, A3/A8, ...
- Software/Updates/Drivers/...
- Digital signatures on legal documents
- Bidding on sugar beets
- Digital/Electronic elections
- Car ignition, updating of pacemakers, etc.

Yet Security is not Achieved

- Viruses/Worms/Trojans/...still exist
- Internet traffic is mostly in the clear
- Most communications is not authenticated
- User authentication is based on PINs and weak passwords
- Most storage is not encrypted
- Data is not shared using the "right tools"
- 40-bit encryption is still around
- Really weak encryption is still around

Motivation of this Talk (part I)

Motivation of this Talk (part I)

- Meet "Matt" — a security engineer
- Matt went to a good school in CS/EE/Math
- Matt took some courses in cryptography/computer security
- He is now working for a respectable company with other security engineers
- Matt knows how to make things secure

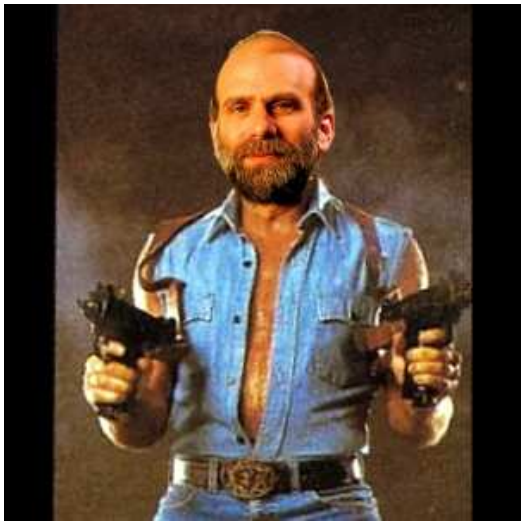
Motivation of this Talk (part II)

Motivation of this Talk (part II)

- Meet "Chris" — a world renowned researcher in cryptography
- Chris has a Ph.D. in cryptography (under the supervision of a distinguished researcher)
- Chris has a position in a leading university
- Chris is involved with the computer security community as well
- Chris is also doing some consulting
- Chris taught Matt...

Motivation of this Talk (part II)

- Meet "Chris" — a world renowned researcher in cryptography
- Chris has a Ph.D. in cryptography (under the supervision of a distinguished researcher)
- Chris has a position in a leading university
- Chris is involved with the computer security community as well
- Chris is also doing some consulting
- Chris taught Matt...



Motivation of this Talk (part III)

- We will cover a few cases of critical failures that could have been prevented if only Matt had listened to Chris
- This is despite Matt's knowledge and training

Motivation of this Talk (part III)

- We will cover a few cases of critical failures that could have been prevented if only Matt had listened to Chris
- This is despite Matt's knowledge and training
- The talk's main message is "You Should Have Known Better"

Motivation of this Talk (part III)

- We will cover a few cases of critical failures that could have been prevented if only Matt had listened to Chris
- This is despite Matt's knowledge and training
- The talk's main message is "You Should Have Known Better"
- Later, we mention a few methods to reduce these issues in the future

The Content Distribution Problem

- The content is to be sold to the consumers, and they are allowed only viewing it

The Content Distribution Problem

- The content is to be sold to the consumers, and they are allowed only viewing it
- Assume Matt's company was hired to fight content-piracy

The Content Distribution Problem

- The content is to be sold to the consumers, and they are allowed only viewing it
- Assume Matt's company was hired to fight content-piracy
- First step in the solution: Encrypt the content on the storage

The Content Distribution Problem

- The content is to be sold to the consumers, and they are allowed only viewing it
- Assume Matt's company was hired to fight content-piracy
- First step in the solution: Encrypt the content on the storage
- Second step: Allow only approved entities to decrypt

The Content Distribution Problem

- The content is to be sold to the consumers, and they are allowed only viewing it
- Assume Matt's company was hired to fight content-piracy
- First step in the solution: Encrypt the content on the storage
- Second step: Allow only approved entities to decrypt
- Third step: Sparkle some key management, traitor tracing schemes, and maybe watermarks

The DVD Case — CSS Algorithm

- DVDs are encrypted using the CSS encryption algorithm
- The CSS scheme is a 40-bit stream cipher
- Each DVD is encrypted using CSS with some random key
- The encryption key is encrypted under various static player keys (for approved players)
- In addition there is a key used for authenticating the reader to the DVD

Recall the DeCSS

- Well, once one of the static keys is found, game is over
- DeCSS just finds keys (40-bit security at its best)
- After decryption the content is no longer protected
- CSS also suffers from design weakness, exhaustive search takes only 2^{16}
- To make things worse, the AACS 128-bit key 09 F9 used for HD DVD and Blu-ray was also found (as well as others)

You Should Have Known Better

- Master keys (and static keys) used to encrypt a lot of data is bad
- 40-bit key is not enough
- Weak cryptography with long keys is not enough as well
- The user can attack the system however she wishes
- Make sure your adversarial model is correct

The MD/SHA Family

- Started with Rivest's MD4 (1990)
- Following a few cryptanalytic attempts, was upgraded to MD5
- MD5, also known to many as **md5sum**, generates tags of 128 bits
- Became very popular given its high speed, alleged security, and lack of true competition...
- Later, it was used as the basis for the SHA-0 and SHA-1 hash functions

The (Quick and Dirty) History of MD5's Security

1993 den Boer & Bosselares: Compression function collision

1996 Dobbertin: Free-start collision

2004 Wang: Practical collision attacks (one hour for a collision)

2005 Klima: Collisions in 8 hours on a laptop

2006 Klima or Stevens: Collisions in a minute

2007 Stevens, Lenstra, de Wegger: Colliding X.509 certificates (same public key, different DN)

2007 Leurent: Extracting passwords from APOP using MD5 Collisions

The (Quick and Dirty) History of MD5's Security

1993 den Boer & Bosselares: Compression function collision

1996 Dobbertin: Free-start collision

2004 Wang: Practical collision attacks (one hour for a collision)

2005 Klima: Collisions in 8 hours on a laptop

2006 Klima or Stevens: Collisions in a minute

2007 Stevens, Lenstra, de Wegger: Colliding X.509 certificates (same public key, different DN)

2007 Leurent: Extracting passwords from APOP using MD5 Collisions

The cryptographic community called for the removal of MD5 from anything important...

And then came Stevens et al.

And then came Stevens et al.

- At CCC08 Stevens et al. reported that they successfully generated a “real-life” certificate
- The attack was based on MD5 collisions, cluster of a PS3s, and the fact that RapidSSL was still using MD5-based signatures

And then came Stevens et al.

- At CCC08 Stevens et al. reported that they successfully generated a "real-life" certificate
- The attack was based on MD5 collisions, cluster of a PS3s, and the fact that RapidSSL was still using MD5-based signatures

You Should Have Known Better!

The Wired Equivalent Protocol (WEP)

- WEP was designed in the late 1990's to have WiFi with the same security as Ethernet
- A static 40-bit key is used in each network
- The key is fed into RC4, along with per-packet 24-bit IV (chosen by the sender)
- Comes with a simple challenge-response protocol for authentication of users
- Also comes with CRC padding on the data, which is also encrypted...

"Non-Cryptanalytic" Problems

1. Short key
2. IV space too small
3. For each IV the same key stream is generated
4. Easy to change the ciphertext to change the plaintext
5. Easy to bypass authentication/integrity check (which actually leaks key stream per IV)

Some Common Sense

- WEP first uses a linear error correction code and then encrypts using the XOR operation
- Thus, violating (one of the) first rule of cryptography:

  "Thou shalt compress, encrypt, and then authenticate. Thou shalt have no other sequences before this"

- In addition, everything is commutative . . .

Some Common Sense

- WEP first uses a linear error correction code and then encrypts using the XOR operation
- Thus, violating (one of the) first rule of cryptography:

  "Thou shalt compress, encrypt, and then authenticate. Thou shalt have no other sequences before this"

- In addition, everything is commutative ...

Some Common Sense

- WEP first uses a linear error correction code and then encrypts using the XOR operation
- Thus, violating (one of the) first rule of cryptography:

  "Thou shalt compress, encrypt, and then authenticate. Thou shalt have no other sequences before this"

- In addition, everything is commutative ...

The Cryptanalytic Problem

- For each key, there are weak IVs, so weak you can find the key [FMS01]
- (One of the nicest examples why related-key attacks are real)
- First attacks required a few million known "packets"
- Quickly implemented (e.g., [SIR01]), and became very popular in software packages
- New attacks emerge now and then
- Best known attack requires only 4,000 "known" packets [SVV11]

And Then Some More Anecdotes

- Shared prime numbers among RSA public keys
- Debian's lack of sufficient entropy
- "hard disk encryption" done using ECB
- GSM/3G security — weak algorithms, encryption after error correction, unauthenticated control channel, etc.
- Using 80's ciphers today — MiFare, KeeLoq, DES(!)
- Making up your own crypto-algorithms
- MD5 **still** used in the context of digital signatures
- Keeping keys in swappable memory (or accessible memory)
- Broken standards/protocols still in use
- ...

And Now For Something Completely Different

The True Motivation of this Talk

- Despite what it seems, not all crypto-related security issues are caused by Matt

The True Motivation of this Talk

- Despite what it seems, not all crypto-related security issues are caused by Matt
- Despite common belief, many are caused by the cryptographers

The True Motivation of this Talk

- Despite what it seems, not all crypto-related security issues are caused by Matt
- Despite common belief, many are caused by the cryptographers
- Some of these problems are the result of the way **cryptographers** think of the problem
- Some of these problems are due to the way the problem is communicated to the cryptographic circles
- Some of it is plain "Cryptographers should have known better"

The RC4 Stream Cipher

- RC4 is a stream cipher designed by Rivest in 1987
- It was part of an RSA library, and its true design was never released

The RC4 Stream Cipher

- RC4 is a stream cipher designed by Rivest in 1987
- It was part of an RSA library, and its true design was never released
- The alleged RC4 was leaked in 1994, and in the response to the attacks on WEP, the design of RC4 was confirmed

The RC4 Stream Cipher

- RC4 is a stream cipher designed by Rivest in 1987
- It was part of an RSA library, and its true design was never released
- The alleged RC4 was leaked in 1994, and in the response to the attacks on WEP, the design of RC4 was confirmed
- This is one of the basic stream ciphers, very efficient in software, and very popular in protocols/products/standards

The RC4 Stream Cipher

- RC4 is a stream cipher designed by Rivest in 1987
- It was part of an RSA library, and its true design was never released
- The alleged RC4 was leaked in 1994, and in the response to the attacks on WEP, the design of RC4 was confirmed
- This is one of the basic stream ciphers, very efficient in software, and very popular in protocols/products/standards
- Went through huge amount of cryptanalytic attacks, no "real" key recovery attack
- Passed huge amounts of statistical tests

The RC4 Stream Cipher

- RC4 is a stream cipher designed by Rivest in 1987
- It was part of an RSA library, and its true design was never released
- The alleged RC4 was leaked in 1994, and in the response to the attacks on WEP, the design of RC4 was confirmed
- This is one of the basic stream ciphers, very efficient in software, and very popular in protocols/products/standards
- Went through huge amount of cryptanalytic attacks, no "real" key recovery attack
- Passed huge amounts of statistical tests
- Common "Folklore": a good cipher to use

Security Issues in RC4

- Surprisingly, RC4's output is biased
- Several well known biases:
  1. Second byte is zero with probability $2/256$ [MS01],
  2. Each output byte is zero with probability slightly more than $1/256$ [MPS11],
  3. Second byte is two with probability significantly lower than $1/256$ [S13],
  4. r'th output byte being $-r$ (with probability slowly decreasing towards $1/256$) [A+13,I+13]
- Lots of research about these biases

Security Issues in RC4

- Surprisingly, RC4's output is biased
- Several well known biases:
  1. Second byte is zero with probability $2/256$ [MS01],
  2. Each output byte is zero with probability slightly more than $1/256$ [MPS11],
  3. Second byte is two with probability significantly lower than $1/256$ [S13],
  4. r'th output byte being $-r$ (with probability slowly decreasing towards $1/256$) [A+13,I+13]
- Lots of research about these biases
- Recently: [A+13] showed the existence of sets of biases allowing retrieving the first 256 bytes of a plaintext encrypted under 2^{32} random RC4 keys.

Security Issues in RC4 (cont.)

- Most of these biases could not have been foreseen
- Most of these biases should have been found before

Security Issues in RC4 (cont.)

- Most of these biases could not have been foreseen
- Most of these biases should have been found before
- However, most statistical tests that were run, took a keystream generated by a single key, and analyzed it

Security Issues in RC4 (cont.)

- Most of these biases could not have been foreseen
- Most of these biases should have been found before
- However, most statistical tests that were run, took a keystream generated by a single key, and analyzed it
- What should have been done — take many key streams generated by many keys

Security Issues in RC4 (cont.)

- Most of these biases could not have been foreseen
- Most of these biases should have been found before
- However, most statistical tests that were run, took a keystream generated by a single key, and analyzed it
- What should have been done — take many key streams generated by many keys
- Result: though RC4 is “secure”, using it in any broadcast environment is a bad idea

Security Issues in RC4 (cont.)

- Most of these biases could not have been foreseen
- Most of these biases should have been found before
- However, most statistical tests that were run, took a
  k t
- W ams
  g
- R adcast
  e

The Advanced Encryption Standard

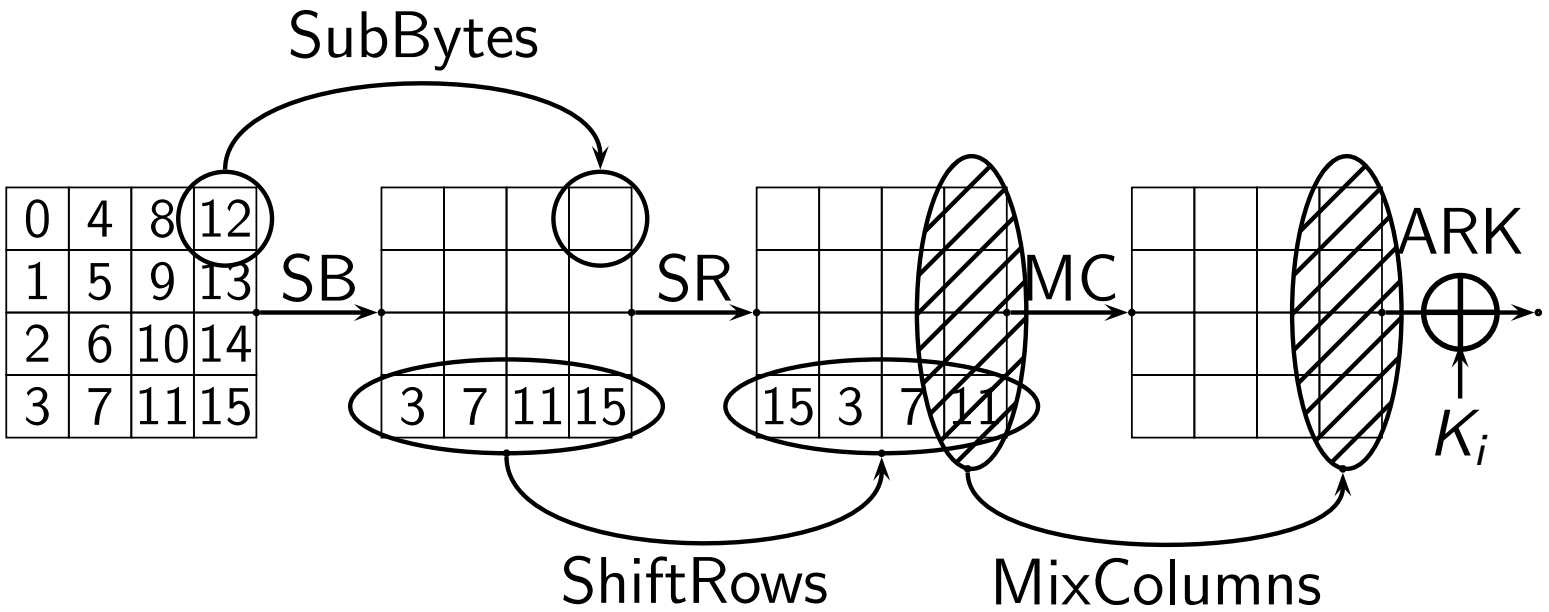
- Rijndael was selected as AES in 2000
- AES security was thoroughly analyzed against:
  - Differential,
  - Linear,
  - Meet in the middle,
  - Algebraic,
  - Impossible differential,
  - Square/Integral/Saturation,
  - Demirci-Selçuk,
  - Boomerang,
  - Related-Key attacks
  - ...

The Advanced Encryption Standard

- Rijndael was selected as AES in 2000
- AES security was thoroughly analyzed against:
  - Differential,
  - Linear,
  - Meet in the middle,
  - Algebraic,
  - Impossible differential,
  - Square/Integral/Saturation,
  - Demirci-Selçuk,
  - Boomerang,
  - Related-Key attacks
  - ...
- Up to a few small issues in very special attack models/small reduction in exhaustive search's complexity — this is the cipher to use

The Advanced Encryption Standard (cont.)

- The cipher has an SP (substitution-permutation) network structure.
- Block size — 128 bits, Key size — 128, 192, or 256 bits.
- Number of rounds depends on the key length (10/12/14, respectively).



The Advanced Encryption Standard (cont.)

- For efficiency reasons, one can implement the SubBytes and the MixColumns operation together using a memory lookup
- There are four 8-bit to 32-bit tables used in most implementations

The Advanced Encryption Standard (cont.)

- For efficiency reasons, one can implement the SubBytes and the MixColumns operation together using a memory lookup
- There are four 8-bit to 32-bit tables used in most implementations
- The last round has no MixColumns
- So there is a fifth 8-bit to 8-bit table

The Advanced Encryption Standard (cont.)

- For efficiency reasons, one can implement the SubBytes and the MixColumns operation together using a memory lookup
- There are four 8-bit to 32-bit tables used in most implementations
- The last round has no MixColumns
- So there is a fifth 8-bit to 8-bit table
- This table is accessed only during the last round of encryption...

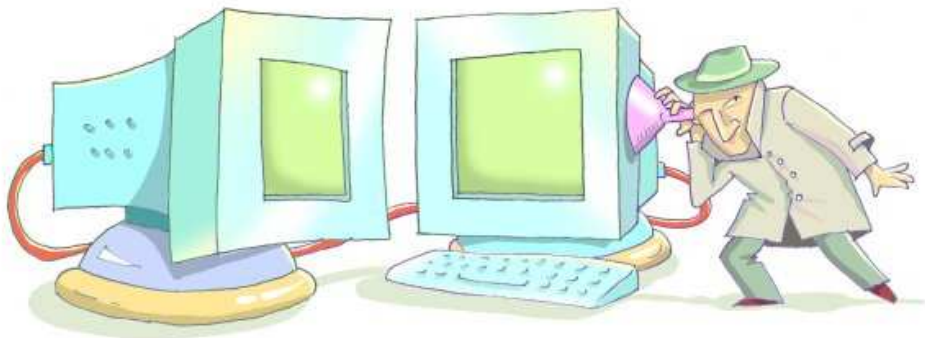
[P02, Ber05, OST06, ...]

Cache Attack on AES (Concept)

1. Flush the cache (filling it with information)
2. Call the encryption process
3. Identify which entries of the fifth table were accessed (time the time needed to access the cache again)

Cache Attack on AES (Concept)

1. Flush the cache (filling it with information)
2. Call the encryption process
3. Identify which entries of the fifth table were accessed (time the time needed to access the cache again)
4. Make Profit



Cache Attacks (and other Software Side Channel Attacks)

- Memory access patterns leak information

Cache Attacks (and other Software Side Channel Attacks)

- Memory access patterns leak information
- Execution times leak information (sometimes can be observed from outside the system)

Cache Attacks (and other Software Side Channel Attacks)

- Memory access patterns leak information
- Execution times leak information (sometimes can be observed from outside the system)
- Branch depending on data controlled by the adversary may change execution times...
- [BT11] a few minutes to recover a machine's OpenSSL ECDSA secret key

Cache Attacks (and other Software Side Channel Attacks)

- Memory access patterns leak information
- Execution times leak information (sometimes can be observed from outside the system)
- Branch depending on data controlled by the adversary may change execution times...
- [BT11] a few minutes to recover a machine's OpenSSL ECDSA secret key **remotely**

Side Channel Attacks

- There are also hardware-based side channel attacks:
  - Power analysis (simple/differential)
  - Template attacks
  - Acoustic attacks
  - Electromagnetic radiation attacks
  - ...

Security engineers should pick not only **secure schemes**, but also **secure implementations!**

Padding in Cryptography

- Textbook RSA:

$$c = m^e \bmod n; sig = m^d \bmod n$$

Padding in Cryptography

- Textbook RSA:

$$c = m^e \bmod n; sig = m^d \bmod n$$

- Using textbook RSA would lead to crypto-abyss! Do not use it!

Padding in Cryptography

- Textbook RSA:

$$c = m^e \bmod n; sig = m^d \bmod n$$

- Using textbook RSA would lead to crypto-abyss! Do not use it!
- Real life RSA-encryption (RSA-OAEP):

$$c = [(m||0||r) \oplus (G(r)||H((m||0) \oplus G(r)))]^e \bmod n$$

- Use similar constructions for RSA-PSS

Padding Attacks

- Due to difference in reactions, one can deduce when the decryption was successful
- This allowed Bleichenbacher to decrypt RSA in SSL using about 10^6 queries
- The problem: implementation first decrypts, then checks padding
- Padding errors are treated differently then "context" errors

Padding Attacks — Symmetric Key

- Similar attacks exist for symmetric-key encryption
- The times are significantly shorter, but are still applicable
- For example, several CBC padding schemes are: one byte of “1”, or two bytes of “2”, or three of “3”, . . .

Padding Attacks — Symmetric Key

- Similar attacks exist for symmetric-key encryption
- The times are significantly shorter, but are still applicable
- For example, several CBC padding schemes are: one byte of "1", or two bytes of "2", or three of "3", ...
- The idea is to manipulate a ciphertext, until the padding is "correct"
- Allows decrypting the last block of an encrypted message [V02]

Padding Attacks — Symmetric Key

- Similar attacks exist for symmetric-key encryption
- The times are significantly shorter, but are still applicable
- For example, several CBC padding schemes are: one byte of "1", or two bytes of "2", or three of "3", ...
- The idea is to manipulate a ciphertext, until the padding is "correct"
- Allows decrypting the last block of an encrypted message [V02]
- Applicable to TLS 1.0, IPsec, SSH, ...

Padding Attacks — Symmetric Key

- Similar attacks exist for symmetric-key encryption
- The times are significantly shorter, but are still applicable
- For example, several CBC padding schemes are: one byte of "1", or two bytes of "2", or three of "3", ...
- The idea is to manipulate a ciphertext, until the padding is "correct"
- Allows decrypting the last block of an encrypted message [V02]
- Applicable to TLS 1.0, IPsec, SSH, ...
- Result: Even when "provably"-secure constructions are used with secure primitives, security flaws exist

Padding Attacks — Symmetric Key

- Similar attacks exist for symmetric-key encryption
- The times are significantly shorter, but are still applicable
- For example, several CBC padding schemes are: one byte of “1”, or two bytes of “2”, or three of “3”, . . .
- The idea is to manipulate a ciphertext, until the padding is
- A
  m
- A
- R are
  u

Cold Boot Attack

- Yesterday Alex showed the cold boot attack
- The attacks use the fact that you can literally “freeze” volatile memory to read its contents later
- As the memory stores the keys, you can read them in the lab afterwards
- Applicable to both public key and secret key primitives

Leakage Resilient Cryptography

- Once the cold boot attack was publicized, cryptographers start to construct leakage-resilient cryptography:
  - Leakage-resilient encryption,
  - Leakage-resilient signatures,
  - Leakage-resilient secure multiparty computation,
- Under different adversarial models:
  - Single leak vs. Continuous leak,
  - Amount of leaked keying material,
  - Different leakage functions,
- Conclusion: We can beat security engineering problems with the right cryptography

In Theory, there is no Difference between Theory and Practice

- [R+11] conducted a thorough analysis of leakage models
- The outcome: basic assumptions of the leakage-resilient cryptography is incorrect for new technologies
- More precisely, in new technologies, independent computations do not cause independent leakage
- In other words, the security assumption is void...

In Theory, there is no Difference between Theory and Practice

- [P+11] conducted a thorough analysis of leakage models
- T lient
  cl
- M
  c
- Ir

When Authentication is not Used

- Encryption does not guarantee security

When Authentication is not Used

- Encryption does not guarantee security
- The lack of authentication may cause security issues:
  - Rouge GPS signals,
  - Allow for attacks such as padding attacks,
  - Replay attacks,
  - The storage device is not always “naive”,
  - ...

When Authentication is not Used

- Encryption does not guarantee security
- The lack of authentication may cause security issues:
  - Rouge GPS signals,
  - Allow for attacks such as padding attacks,
  - Replay attacks,
  - The storage device is not always “naive”,
  - ...
- But authentication is cheap!

Better Future — Today

Better Future — Today

- More emphasis in cryptographic research on real life

Better Future — Today

- More emphasis in cryptographic research on real life
- More collaboration between cryptographers and the security industry (e.g., standardization bodies)

Better Future — Today

- More emphasis in cryptographic research on real life
- More collaboration between cryptographers and the security industry (e.g., standardization bodies)
- Better sharing mechanisms for dissemination of "cryptographic" knowledge

Better Future — Today

- More emphasis in cryptographic research on real life
- More collaboration between cryptographers and the security industry (e.g., standardization bodies)
- Better sharing mechanisms for dissemination of "cryptographic" knowledge
- Cryptographers may need to do the actual "coding" / "construction" of systems

Better Future — Today

- More emphasis in cryptographic research on real life
- More collaboration between cryptographers and the security industry (e.g., standardization bodies)
- Better sharing mechanisms for dissemination of "cryptographic" knowledge
- Cryptographers may need to do the actual "coding" / "construction" of systems
- Generally — security engineers and cryptographers should meet more often

The Real Thing

- Fault-tolerant design
- Key-agility design (especially key sizes)
- Crypto-algorithm-agility
- Better generic libraries (software/hardware)

Some of the Steps

- The CAESER competition (for authenticated-encryption solutions)
- The NaCl library
- Patent-free inclination in standards/protocols/research
- Series of Crypto-in-real-world events:
  - Real-World Cryptography (2011, 2013, 2014, ...)
  - International State of the Art in Cryptography and Computer Security (Dagstuhl 2011, ASIACRYPT 2012, EUROCRYPT 2013, ASIACRYPT 2013, ...)
  - CHES & CRYPTO meet (once evrey three years)
  - ...

Some of the Steps

- The CAESER competition (for authenticated-encryption solutions)
- The NaCl library
- Patent-free inclination in standards/protocols/research
- Series of Crypto-in-real-world events:
  - Real-World Cryptography (2011, 2013, 2014, ...)
  - International State of the Art in Cryptography and Computer Security (Dagstuhl 2011, ASIACRYPT 2012, EUROCRYPT 2013, ASIACRYPT 2013, ...)
  - CHES & CRYPTO meet (once evrey three years)
  - ...
- More realistic attack models

Questions?

Thank you very much for your attention!