Royal Holloway University of London — Information Security Group

Does **provably secure** cryptography guarantee **practical** security?

Chris Mitchell
Royal Holloway
`www.isg.rhul.ac.uk/~cjm`

1

Royal Holloway University of London — Information Security Group

Agenda

1. What is provable security?
2. Benefits
3. Limitations
4. Case studies
5. Answering the question ...

2

Royal Holloway University of London — Information Security Group

Agenda

1. What is provable security?
2. Benefits
3. Limitations
4. Case studies
5. Answering the question ...

3

Royal Holloway University of London — Information Security Group

History I

- Until relatively recently, cryptographic algorithms (with the exception of the one-time pad) were designed and used according to purely heuristic principles.
- That is, algorithm designers used their experience of breaking algorithms to try to come up with schemes which resisted any known cryptanalysis methods.

4

Royal Holloway University of London — Information Security Group

History II

- The same applied (and still often applies) to systems using cryptography.
- Designers would put complex systems together in the knowledge of a range of attack techniques.
- A system would be deemed secure if an expert (or, preferably, experts) could not find any way of attacking it.

5

Royal Holloway University of London — Information Security Group

Consequences

- This essentially craft-based approach has not been a disaster (many algorithms and systems designed this way have proved very resilient).
- However, it has also yielded a large number of algorithms and systems which have proved very simple to break.
- Indeed, sometimes the ‘experts’ were not as knowledgeable as they might have been.

6

From craft to science

- Several parallel threads of work have emerged, trying to take cryptography from a *craft* to a *science*.
- There are three rigorous approaches of particular importance:
  - logic based cryptographic protocol analysis;
  - information-theoretic security;
  - complexity-theoretic security (or 'provable security').

Logic based approaches

- Since the ground-breaking 1988 paper of Burrows, Abadi and Needham (*A logic of authentication*), a wide range of efforts have been made to use logic to reason about security protocols.
- These logics typically make very (unrealistically) strong assumptions about the cryptographic algorithms employed.
- Despite this, such work has proved very useful in revealing unsuspected flaws in a wide range of protocols.

Information-theoretic security

- Information theoretic security dates back to seminal papers of Shannon, published in the late 1940s.
- An algorithm is information-theoretically secure if, no matter what computational resources are available to the cryptanalyst, the algorithm cannot be broken.
- Such schemes (such as the one-time pad) require large amounts of 'one time' secret keying material, and have somewhat limited practicality.

Complexity-theoretic security I

- Complexity-theoretic security as it is currently constituted dates back only to the early-mid 1990s, and key papers by Bellare & Rogaway (1994/95).
- The roots of the approach, however, go back to Rabin (1978), Goldwasser & Micali (1982), and Goldwasser, Micali & Rivest (1984).
- It is designed to enable rigorous analyses of cryptosystems of the type that have been commonly employed for centuries (in particular, systems that use 'short' keys).
- It involves proving that if a particular algorithm (or system) can be broken, then a problem believed to be hard can be solved.

Complexity-theoretic security II

- More specifically, it involves showing that:
  - **if**: an algorithm exists to solve an instance of a cryptosystem (with security parameter s) in time $f(s)$;
  - **then**: an algorithm exists to solve a problem believed to be hard (e.g. the discrete logarithm problem) in time $g(f(s))$, for some polynomial g (where s is also a parameter for the hard problem).
- I.e., if we can break the cryptosystem in polynomial time, then we can solve the hard problem in polynomial time.

Parameterisation

- Complexity theory involves reasoning about the work involved in solving *parameterised* problems of varying size.
- Thus, when applied to cryptography, we need to consider cryptosystems of varying size (e.g. for RSA, the length of the primes) – this size is captured by the security parameter.

Royal Holloway University of London — Information Security Group

Reductions

- The polynomial *g* is itself of interest.
- That is, the lower the degree of *g*, the **tighter the reduction** from one problem to the other.
- That is, if we can ensure *g* has small degree, then we know that the security of the cryptosystem is closely related to the complexity of the hard problem.

13

Royal Holloway University of London — Information Security Group

Cryptographic primitives

- The information theoretic and complexity theoretic approaches both apply to cryptographic primitives (such as encryption algorithms and digital signature schemes).
- However the logic approaches do not address crypto-primitives.

14

Royal Holloway University of London — Information Security Group

Cryptographic protocols

- Typically, either complexity-theoretic or logic based approaches are applied to protocols/systems using cryptography.
- Large systems (e.g. the SET e-commerce protocols and certain WS protocols) have been analysed using logic-based systems.
- Complexity theory approach mainly applied to 'generic' protocols.

15

Royal Holloway University of London — Information Security Group

Complexity theory versus logic

- Logic based approach advantages:
  - automated proofs can be generated (using theorem provers) – much less chance of errors in proofs;
  - larger systems can be reasoned about.
- Complexity theory approach advantages:
  - makes realistic assumptions about cryptographic algorithms.
- Main focus of this talk is the complexity theory approach (favoured by the crypto community), i.e. **provable security**.

16

Royal Holloway University of London — Information Security Group

Agenda

1. What is provable security?
2. Benefits
3. Limitations
4. Case studies
5. Answering the question …

17

Royal Holloway University of London — Information Security Group

Why provable security?

- Gives us greater confidence that cryptographic primitives will not suffer from simple attacks.
- Also enables us to develop greater confidence in the design of cryptographic protocols.
- Has given us new insights into best use of cryptography (e.g. never encrypt without also integrity-protecting).

18

Why use provably secure crypto?

- There is no excuse not to use provably secure crypto-primitives.
- Typically there is a minimal performance hit, and there are growing number of examples of systems which have been broken because the designers ignored the advice provided by provable security – e.g. encryption-only IPsec.
- There are now many standards containing provably secure primitives.

19

Royal Holloway University of London Information Security Group

Basic primitives – asymmetric I

- In asymmetric crypto case, theory starts from assumption that certain problems are hard, e.g.:
  - **RSA problem**: if $n=pq$ (where p and q are large secret primes), then, given a random $c < n$, find x such that: $x^e \bmod n = c$ (where e and n are public).
  - **Diffie-Hellman problem**: if p is a large prime and g has multiplicative order q modulo p (q a large prime, p and g public), then, given ($g^a \bmod p$) and ($g^b \bmod p$) for random secret a and b, find: ($g^{ab} \bmod p$).

20

Royal Holloway University of London Information Security Group

Basic primitives – asymmetric II

- If the **factorisation problem** (given $n=pq$, find p and q) is hard then the RSA problem is hard, but not necessarily vice versa.
- If the **discrete logarithm problem** (given $g^x \bmod p$, find x) is hard then the DH problem is hard, but not necessarily vice versa.
- A host of other related problems are used as the basis of cryptosystems.

21

Royal Holloway University of London Information Security Group

Compound primitives – asymmetric I

- The main goal of provable security is to construct cryptosystems (i.e. complete encryption schemes, signatures schemes, key establishment schemes, etc.) whose security is as good as a specific problem.
- To define security we need to first define an attack model, which will be specific to the type of primitive.
- That is, what can the attacker know?

22

Royal Holloway University of London Information Security Group

Compound primitives – asymmetric II

- For example, for encryption, the attack model enables the attacker to learn the ciphertexts for chosen plaintexts (and vice versa, except for the target ciphertext).
- Security means that, given a target ciphertext, if an algorithm exists for finding the plaintext, then an algorithm can be constructed to break a specific hard problem (with similar complexity).

23

Royal Holloway University of London Information Security Group

Basic primitives – symmetric

- The problems on which provably secure symmetric schemes are based are much less ‘clean’.
- Typically, these primitives are functions such as block ciphers, or round-functions of hash-functions.
- What is there to prove in this case?

24

Compound primitives - asymmetric

- Functions which are proved secure include:
  - block cipher modes of operation (for encryption), e.g. CBC;
  - block cipher modes for computing MACs (typically CBC-MACs), e.g. OMAC/CMAC;
  - Combined confidentiality/integrity modes using block ciphers, such as OCB, CCM or EAX;
  - complete hash-functions using round-functions.

Protocol design

- Provable security techniques have also been applied to general purpose security protocols, e.g. authentication and authenticated key establishment protocols.
- Less commonly applied to specific application protocols, as specific security models need to be devised for each application environment.

Case study – asymmetric crypto

- Encrypting using ‘naive’ RSA, i.e. by encrypting a message m as (m^e mod n) is **not** as difficult to break as solving the RSA problem.
- Need to apply an appropriate randomising function to m before exponentiating.
- Similarly, naive hash-based RSA signatures are also suboptimal.

Case study – symmetric crypto

- Even though encryption modes such as CBC can be proved to have desirable properties (assuming block cipher ‘ideal’), using CBC mode on its own is suboptimal.
- To achieve provable security in most robust attack model, must also integrity-protect.
- This theoretical result translates into real attacks on systems, as is now well-documented.

Agenda

1. What is provable security?
2. Benefits
3. Limitations
4. Case studies
5. Answering the question ...

Is it all over?

- Does the advent of provable security mean that we can all pack up and go home?
- Well, in general, no – although we do seem to have a very nice collection of seemingly robust asymmetric crypto-primitives.
- However, problems remain, and in the reminder of the talk we briefly look at some of the limitations of the current state of the art.

Quality of proofs

- There are huge problems with the quality of many (most?) of the proofs of security in the literature.
- These problems arise for a variety of reasons, including:
  - lack of **space** – the reliance on conference publications limits page length;
  - lack of **time** – papers are prepared in a huge rush, and the refereeing process is typically minimal;
  - lack of **expertise** – many/most proofs are written by authors who do not have rigorous mathematical training.

31

Underlying hard problems I

- The 'hard problems' on which the proofs of security for **asymmetric schemes** are based are often (usually?) not the 'standard' hard problems studied in complexity theory.
- Indeed, there seem to be almost as many problems on which schemes are built as there are schemes.

32

Underlying hard problems II

- This mushrooming of the problem base is rapidly becoming a subject of study in its own right.
- That is, it has become of interest to know the relationships between members of classes of hard problems, and to find low complexity 'reductions' from one version of a problem to another.

33

Underlying hard problems III

- For symmetric crypto, the issues become even more serious.
- In analyses (e.g. of modes of operation) the 'real' components, e.g. block ciphers, are replaced with idealised components, typically involving random families of permutations.
- This is essential since, apart from anything else, real-life block ciphers are not arbitrarily parameterisable.
- However, real-life block ciphers are not the same as the idealised components, raising the possibility of cipher-specific attacks on modes of operation; for example, it has recently been shown that AES-256 cannot be treated as an ideal cipher in complexity-theoretic proofs.

34

Underlying hard problems IV

- One issue not so far mentioned is modelling the use of cryptographic hash-functions in cryptographic schemes (e.g. digital signatures).
- Many proofs model these as **random oracles** – i.e. functions that return a random output (except, given the same input twice, they give the same output).
- This is not totally satisfactory, since there are (artificial) schemes which can be proved secure in the random oracle model and which can be shown to be insecure if the random oracle is replaced with any real-life hash-function.

35

Difficulties with threat models I

- The threat models for the 'basic' set of crypto-primitives are fairly well-established.
- However, models for protocols, in particular application-specific protocols, are more problematic, since they need to be devised to take into account the detailed properties of the application.

36

Difficulties with threat models II

- In the case of real-world protocols, particular problems can be caused by protocol errors.
- These can give information to a cryptanalyst in ways which may not be caught by the threat model.
- These may, in turn, invalidate the security proof (in the sense that the protocol may have unexpected vulnerabilities).

37

Agenda

1. What is provable security?
2. Benefits
3. Limitations
4. Case studies
5. Answering the question ...

38

Examples

- We briefly review two real-life cases of problems which have arisen with schemes believed to be ‘provably secure’.
- We give examples of a failed proof and an inadequate threat model.
- These are probably only the tip of a very large iceberg.
- Indeed, we have chosen as examples very well-known work by major figures in the field.

39

OAEP – background

- Optimal Asymmetric Encryption Padding (OAEP) is a means of converting the RSA primitive into a robust encryption scheme.
- It is due to Bellare and Rogaway (1994).
- Bellare and Rogaway ‘proved’ OAEP to be secure (using the random oracle model) against the most challenging threat model for encryption schemes, known as IND-CCA2.
- Specifically, they reduced breaking RSA-OAEP to the RSA problem.

40

OAEP – deployment

- OAEP was one of the earliest public key encryption schemes with a proof of security, and was adopted in the SET (Secure Electronic Transactions) e-commerce protocol.
- SET subsequently failed for commercial/business reasons, although this was not due to the adoption of OAEP!
- Had SET succeeded, it would have protected the security of all Internet-based credit/debit transactions – a big deal!

41

OAEP – problems

- In 2001, Shoup showed that the original security proof was flawed.
- Fujisaki, Okamoto, Pointcheval and Stern showed in the same year that OAEP was, after all, secure, although perhaps more by accident than design!
- However, the new proof does not have a tight reduction.
- Also in 2001, Manger showed that the IND-CCA2 security of OAEP can easily be undermined by error messages, depending on the system in which it is implemented (and issues with complex threat models for real-life applications)

42

OAEP – the future

- OAEP has been standardised, along with other provably secure encryption schemes (ISO/IEC 18033-2).
- Other RSA-based encryption schemes are now known, some of which are included in the ISO standard.
- These are both provably secure and have tighter reductions that can be achieved with OAEP.
- OAEP standardised primarily for legacy reasons.

43

Royal Holloway University of London Information Security Group

OAEP – analysis

- The history of OAEP shows that even proofs written by top experts can have flaws that remain undetected for years.
- These top experts helped to invent the provable security paradigm.
- If they can get it wrong, what chance for the many unverified ‘proofs’ in the literature?

44

Royal Holloway University of London Information Security Group

SSH – background

- Secure SHell (SSH) provides a secure communications channel.
- Latest version is SSHv2.
- The SSH Binary Packet protocol (BPP), responsible for providing data integrity and confidentiality, has been proven secure by Bellare, Kohno and Namprempre (2004).

45

Royal Holloway University of London Information Security Group

SSH – deployment

- It is widely used across the Internet to support secure remote logins to servers.
- SSH has also become a general purpose tool for securing Internet traffic.
- It is thus a very important real-life protocol for the security of data sent across the Internet.

46

Royal Holloway University of London Information Security Group

SSH – problems

- However, Albrecht, Paterson and Watson (2009) showed serious flaws in SSH.
- By injecting carefully constructed modified data into an SSH channel, plaintext can in some cases be recovered by monitoring error messages.
- Proof of concept implementations of the attack have developed which work against (widely deployed) OpenSSH.

47

Royal Holloway University of London Information Security Group

SSH – analysis

- How can this be?
- The security proofs appear sound
- Well, the problem lies in the threat model.
- The threat model only took account of one possible error message generated by a (legitimate) decrypter.
- In practice, by feeding data to a decrypter in stages, the point at which an error occurs can be detected, thus providing information to an attacker.

48

SSH – the future

- Fortunately, the problem would appear to have been solved by preventing use of CBC mode encryption (and using CTR mode instead).
- This does not repair the security proof, but there are good reasons to believe that no more attacks are possible, at least not of this general type.

Agenda

1. What is provable security?
2. Benefits
3. Limitations
4. Case studies
5. Answering the question ...

Is it worth bothering with provable security?

- **Yes!**
- Although SSH and OAEP had problems, it is certainly better to adopt schemes with security proofs than not.
- There are huge numbers of examples of very badly failed schemes in the past.
- Sadly, the literature of badly designed and unproven cryptosystems is still growing rapidly.

What can we do?

- How can we avoid the problems described, notably with:
  - dodgy proofs;
  - reliance on hard problems which may not always be hard;
  - inappropriate threat models.

It is Mathematics

- One huge problem with the standard of proofs is the way that most material is published.
- Refereeing of conference papers does not allow close scrutiny of proofs.
- Perhaps one underlying problem is the huge volume of weak publications.
- Another problem is the fact that 'proof sketches' and incomplete proofs are widely regarded as acceptable – publishing full proofs in unrefereed e-prints is not a substitute for careful review.

Improving proof quality

- Those designing cryptosystems should also take on the responsibility of providing rigorous proofs, if necessary collaborating more widely to get the job done.
- Those editing journals (and conference proceedings) should simply reject cryptography papers either without proofs, or with proofs not properly constructed.

Royal Holloway University of London | Information Security Group

Care with models

- Perhaps the most difficult problem is formulating threat models that really take into account all aspects of a real-world application.
- Not clear what the solution is – except to work at it, and make sure everyone is aware of the problems caused by error conditions.

55

Royal Holloway University of London | Information Security Group

Future of symmetric crypto

- Currently, symmetric cryptography still relies on heuristic techniques to design fundamental building blocks (block ciphers, hash functions, etc.).
- An end to this situation is not even in sight.
- Perhaps this is the real challenge for the future of crypto ...

56

Royal Holloway University of London | Information Security Group

Thank you ...

- Many thanks to the conference organisers for allowing me to share my thoughts with you.
- Many thanks also to Kenny Paterson for a number of very helpful comments on this talk.
- **Questions?**

I am always happy to respond to questions by email, at: me@chrismitchell,net

57