Another Look at Provable Security

Alfred Menezes

(joint work with Neal Koblitz)

The need for provable security

- Since the mid 1970's, cryptographic protocols for achieving various goals have been proposed at a furious rate.
- It is very desirable to obtain mathematically rigorous proofs that protocols meet their goals under some plausible assumptions.
  - If nothing else, the desire for rigour encourages researchers to carefully define their security notions and precisely state their assumptions.
- Security proofs generally take the form of a reduction:
  - Argue that if the protocol can be broken, then a (reasonable) mathematical assumption is invalid.

Basic Rabin public-key encryption scheme

- First provably secure cryptographic scheme (1979).

Basic Rabin public-key encryption scheme

- First provably secure cryptographic scheme (1979).
- Public key: n
- Private key: p and q (where $n = pq$)
- Encryption: $c = m^2 \bmod n$
- Decryption: Find the 4 square roots of c, and select the appropriate one

Basic Rabin public-key encryption scheme

- First provably secure cryptographic scheme (1979).
- Public key: n
- Private key: p and q (where $n = pq$)
- Encryption: $c = m^2 \bmod n$
- Decryption: Find the 4 square roots of c, and select the appropriate one

- Security proof:

  1. SQUARE-ROOTS $\leq_P$ FACTOR-n

Basic Rabin public-key encryption scheme

- First provably secure cryptographic scheme (1979).
- Public key: n
- Private key: p and q (where $n = pq$)
- Encryption: $c = m^2 \bmod n$
- Decryption: Find the 4 square roots of c, and select the appropriate one

- Security proof:
  1. SQUARE-ROOTS $\leq_P$ FACTOR-n
  2. FACTOR-$n \leq_P$ SQUARE-ROOTS

Very brief history of provable security

- Goldwasser & Micali (1982): Probabilistic encryption
  - Semantic security, indistinguishability

Very brief history of provable security

- Goldwasser & Micali (1982): Probabilistic encryption
  - Semantic security, indistinguishability
- (1980's): Asymptotic security analysis
  - Goldreich: "Claims of plausibility"

Very brief history of provable security

- Goldwasser & Micali (1982): Probabilistic encryption
  - Semantic security, indistinguishability
- (1980's): Asymptotic security analysis
  - Goldreich: "Claims of plausibility"
- Bellare & Rogaway (1993):
  Practice-oriented provable security
  - Prove the security of efficient protocols
  - Exact (non-asymptotic) security analysis
  - Random oracle model

Very brief history of provable security

- Goldwasser & Micali (1982): Probabilistic encryption
  - Semantic security, indistinguishability
- (1980's): Asymptotic security analysis
  - Goldreich: "Claims of plausibility"
- Bellare & Rogaway (1993):
  Practice-oriented provable security
  - Prove the security of efficient protocols
  - Exact (non-asymptotic) security analysis
  - Random oracle model

However, it is not always clear what these proofs *really* mean in practice.

RSA key generation

Alice does the following:

1. Select primes p and q of the same bitlength.
2. Compute $n = pq$ and $\phi = (p-1)(q-1)$.
3. Select arbitrary e, $1 < e < \phi$, with $\gcd(e, \phi) = 1$.
4. Compute $d = e^{-1} \bmod \phi$.

Alice's public key is (n, e); her private key is d.

Computing d from (n, e) is equivalent to factoring n.

Full-Domain-Hash RSA signature scheme

Signature generation: To sign $m \in \{0,1\}^*$, Alice does:

1. Compute $h = H(m)$, where $H : \{0,1\}^* \rightarrow [0, n-1]$ is a (public) hash function.
2. Compute $s = h^d \bmod n$.

Alice's signature on m is s.

Signature verification: To verify, Bob does:

1. Obtain Alice's public key (n, e).
2. Compute $h = H(m)$ and verify that $h = s^e \bmod n$.

Full-Domain-Hash RSA signature scheme

Signature generation: To sign $m \in \{0,1\}^*$, Alice does:

1. Compute $h = H(m)$, where $H : \{0,1\}^* \to [0, n-1]$ is a (public) hash function.
2. Compute $s = h^d \bmod n$.

Alice's signature on m is s.

Signature verification: To verify, Bob does:

1. Obtain Alice's public key (n, e).
2. Compute $h = H(m)$ and verify that $h = s^e \bmod n$.

Question: Is RSA-FDH secure?

Security definition

Definition: (Goldwasser, Micali & Rivest; 1984)
A signature scheme is secure if a computationally bounded attacker who has access to a signing oracle is unable (with non-negligible probability) to obtain a valid signature for any message that it did not previously present to the oracle.

Necessary conditions for security of RSA-FDH

1. The RSA problem should be intractable:
   Given n, e, h, find s such that $h = s^e \bmod n$.
   - Clearly RSA $\leq_P$ FACTOR-n.
   - Open Question: FACTOR-$n \leq_P$ RSA?
   - $e = 3$ is commonly used in practice.
   - Boneh and Venkatesan (1998) proved that, if FACTOR-$n \leq_P$ RSA-3 where the reduction algorithm uses only algebraic operations, then FACTOR-n is in P.
   - Nevertheless, we assume that RSA-3 and RSA are as hard as FACTOR-n in practice.

Necessary conditions for security of RSA-FDH

2. H should be preimage resistant:
   - Otherwise, an attacker could first select s, then compute $h = s^e \bmod n$, and finally select m with $H(m) = h$.

Necessary conditions for security of RSA-FDH

2. H should be preimage resistant:
   - Otherwise, an attacker could first select s, then compute $h = s^e \bmod n$, and finally select m with $H(m) = h$.

3. H should be collision resistant:
   - Otherwise, an attacker could find m_1, m_2 with $H(m_1) = H(m_2)$ and induce Alice to sign m_1, thereby obtaining Alice's signature on m_2.

Necessary conditions for security of RSA-FDH

2. H should be preimage resistant:
   - Otherwise, an attacker could first select s, then compute $h = s^e \bmod n$, and finally select m with $H(m) = h$.

3. H should be collision resistant:
   - Otherwise, an attacker could find m_1, m_2 with $H(m_1) = H(m_2)$ and induce Alice to sign m_1, thereby obtaining Alice's signature on m_2.

Question: Are these conditions sufficient?

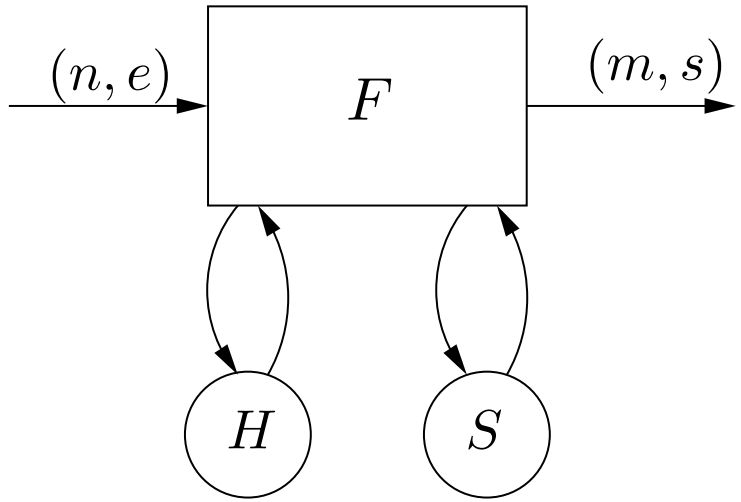
Random oracle model

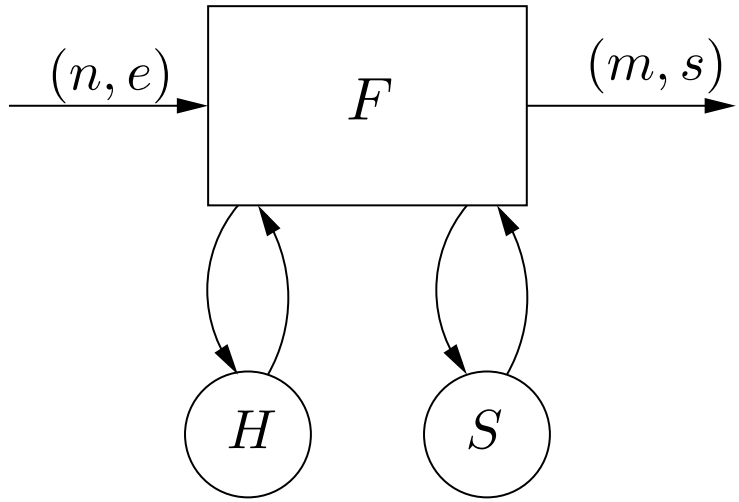
- The hash function H is modeled as a (public) random function.
- The adversary's probability of success is assessed over all possible hash functions.
- This is the random oracle assumption.

Security proof

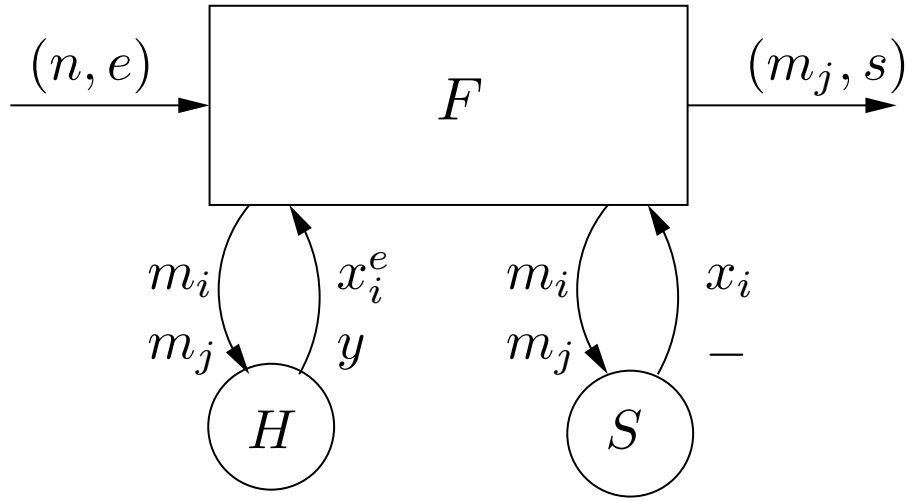
Theorem (Bellare & Rogaway; 1993) RSA-FDH is a secure signature scheme in the random oracle model under the assumption that the RSA problem is intractable.
Proof:

1. Let F be an attacker that breaks RSA-FDH. F can make calls to two subroutines, a hash oracle H and a signing oracle S. At the end of its operation, F produces (with non-negligible probability) a signature for a message not presented to the signing oracle. We show how such a program F can be used to solve the RSA problem.



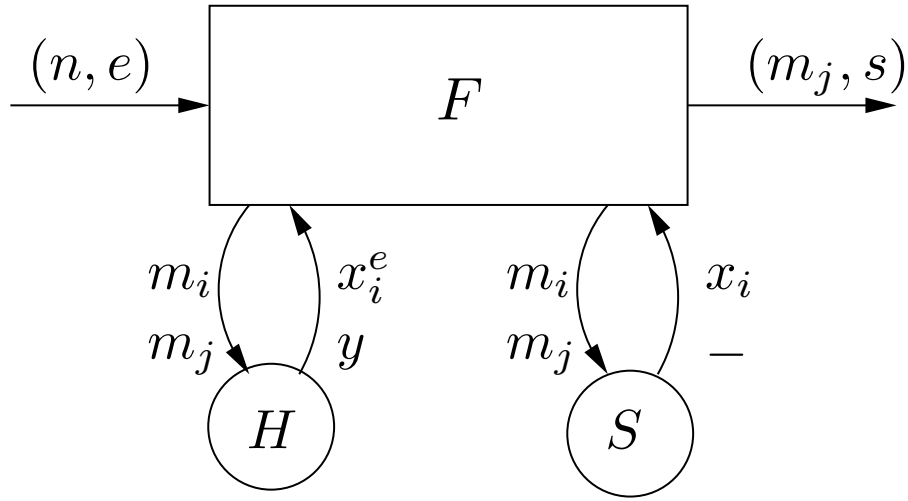


2. Suppose that we are given an instance (n, e, y) of the RSA problem. Our task is to find x such that $y \equiv x^e \pmod{n}$.
3. We make two assumptions about F's operation:
   a) Before querying S with m, F always queries H with m.
   b) F makes (at most) q (distinct) H-queries, $m_1, m_2, \ldots, m_q$.
   c) F outputs a valid signature on one of the m_i's.
4. We select a random index $j \in [1, q]$.
5. We run F with input (n, e) and wait for its queries.



6. For all H-queries except for the jth one, we select a random $x_i \in [0, n-1]$ and respond with $H(m_i) = x_i^e \bmod n$.
   For the jth H-query, we respond with $H(m_j) = y$.

7. If an S-query on m_i is issued (where $i \neq j$) we respond with x_i. (Note: this is a valid signature.)
   If an S-query on m_j is issued, we give up (restart F and select a new j).



8. Suppose that F outputs a valid signature s on m_j. Then $s^e \equiv H(m_j) \equiv y \pmod{n}$, and so $x = s$ is the solution to our RSA problem instance.
   If F does not output a valid signature on m_j, we restart F (and select a new j).

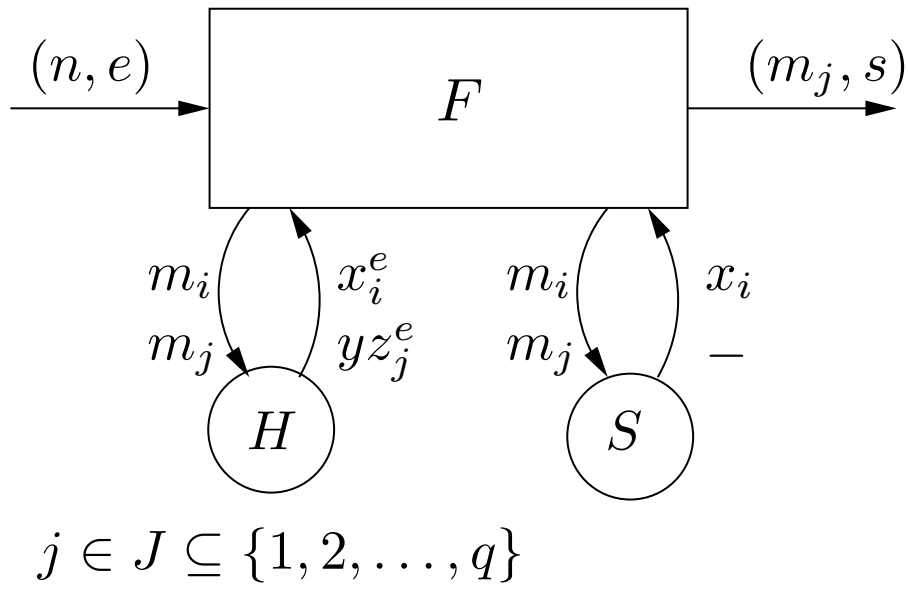
9. If we repeat this procedure k times, the probability that every single time we fail is at most $(1 - 1/q)^k$. $\square$

Tightness of the reduction

- The forgery program F would have to be used roughly q times in order to find the desired e-th root of y.
  - This is a highly non-tight reduction.
- Suppose that n is a 1024-bit integer.
- The NFS takes about 2^{80} steps to factor n.
  - Suppose that the RSA problem cannot be solved in fewer than 2^{80} steps.
- Suppose the forger can make at most $q = 2^{70}$ H queries.
- Then the Bellare-Rogaway proof says that a successful forger must require time at least 2^{10}.
- So, if we desire the assurance that any forger must take time at least 2^{80} then we need to select n so that factoring takes time at least 2^{150} steps.
  - That is, we should use a $\approx$ 4000-bit modulus n.

A tighter reduction

- In 2000, Coron gave a different reduction which lowered the number to F-invocations to q_s (where q_s is a bound on the number of signature queries).



- Coron (2001) proved that no “tighter” reduction is possible.

Practical interpretation

- Suppose that n is a 1024-bit integer.
- Suppose that the forger can make at most $q_s = 2^{20}$ signature queries.
- Then Coron's proof says that a successful forger must require time at least 2^{60}.
- So, if we desire the assurance that any forger must take time at least 2^{80} then we need to select n so that factoring takes time at least 2^{100} steps.
  - That is, we should use a 1500-bit modulus n.

RSA-PSS (simplified)

(Bellare & Rogaway, 1996)

Signature generation: To sign $m \in \{0,1\}^*$, Alice does:

1. Select a random bit string r.
2. Compute $h = H(m, r)$, where $H : \{0,1\}^* \rightarrow [0, n-1]$ is a (public) hash function.
3. Compute $s = h^d \bmod n$.

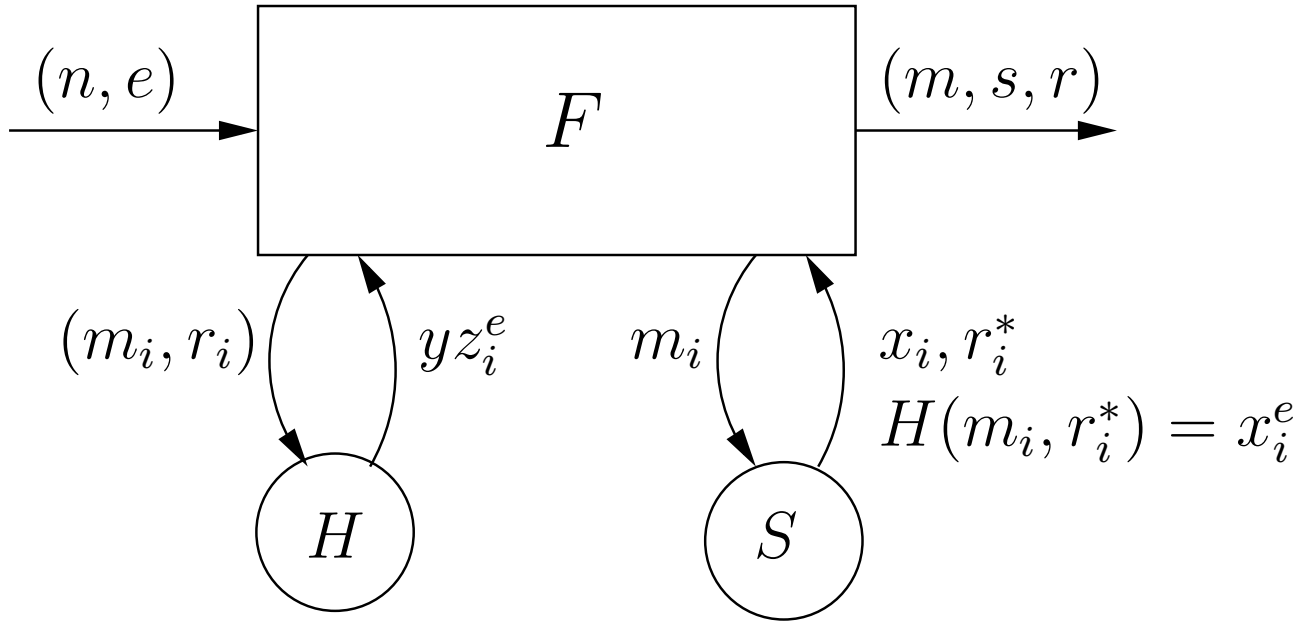
Alice's signature on m is (s, r).

Signature verification: To verify, Bob does:

1. Obtain Alice's public key (n, e).
2. Compute $h = H(m, r)$ and verify that $h = s^e \bmod n$.

Security proof

The security of RSA-PSS can be tightly related to the hardness of the RSA problem.



FDH versus PSS

- Both RSA-FDH and RSA-PSS have security proofs under the same assumptions.
- The signing and verification procedures are equally fast.

- Advantage of RSA-FDH: No random numbers are needed.
- Advantage of RSA-PSS: Has a tight reduction.

Standards have been favouring RSA-PSS.

FDH or PSS?

- The security of RSA-FDH is tightly related to the hardness of the following problem:
  - RSA1: Given (n, e), and a set of q values y_i randomly chosen from $[0, n-1]$, you are permitted at any time to select up to q_s of those y_i for which you will be given solutions x_i to $x_i^e \equiv y_i \pmod{n}$. You must produce a solution $x_i^e \equiv y_i \pmod{n}$ for one of the remaining y_i.
- Even though there is no tight reduction from RSA to RSA1, it is reasonable to conjecture that RSA and RSA1 are equivalent in practice — no one will ever be able to find a solution to RSA1 without being able to solve RSA in essentially the same amount of time.
- One reasonable conclusion: The lack of a tight security reduction for RSA-FDH is not a concern.

RSA-KW (Katz-Wang, 2003)

Key generation: Alice selects a (secret) random bit string R.

Signature generation: To sign $m \in \{0,1\}^*$, Alice does:

1. Compute the bit $b = H_2(m, R)$.
2. Compute $h = H(m, b)$.
3. Compute $s = h^d \bmod n$.

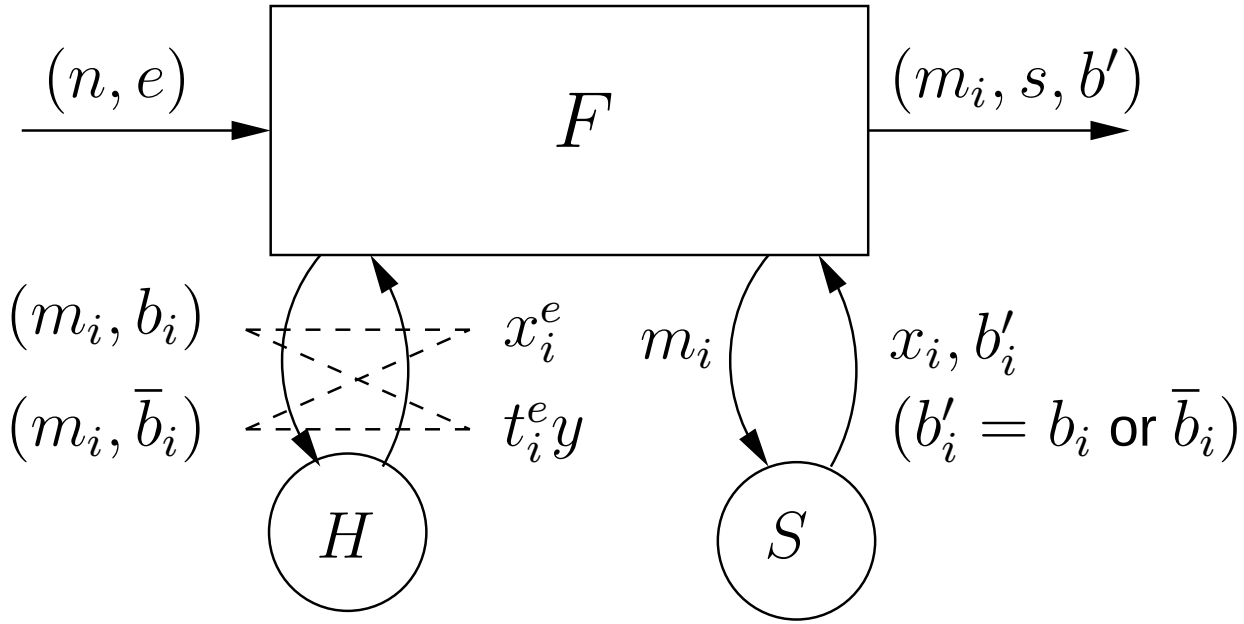
Alice's signature on m is (s, b).

Signature verification: To verify, Bob does:

1. Obtain Alice's public key (n, e).
2. Compute $h = H(m, b)$ and verify that $h = s^e \bmod n$.

Security proof

The security of RSA-KW can be tightly related to the hardness of the RSA problem.



FDH or KW?

- The security of RSA-KW is also tightly related to the hardness of the following problem:
  RSA2: Given (n, e), and a set of q pairs of values (y_i, z_i) chosen at random from $[0, n-1]$, you are permitted at any time to select up to q_s of those pairs for which you will be given the e-th root modulo n of exactly one (randomly selected) element of the pair. You must produce an e-th root of either element in one of the remaining pairs.
- Coron's result implies that there is no tight reduction from the RSA2 problem to the RSA1 problem.
- However, it seems very unlikely that RSA1 would be easier to solve in practice than RSA2.

Random oracle assumption

- In practice, H is not a random function, so the security proof is no longer valid.
- Nevertheless, the security proof does guarantee security against attackers who do not exploit any property whatsoever of the hash function H.
- Several researchers have designed protocols that are provably secure in the random oracle model, but provably insecure whenever the random oracle is replaced by a real hash function.
- However, these protocols are very contrived and arguably support the random oracle model.

Canetti-Goldreich-Halevi example

- Suppose signature scheme S is secure in the random oracle model (with random function H).
- Make the following modification to S to obtain a scheme S' that is also secure in the random oracle model:
  - If $H(m) =$ SHA-1(0) then include the private key in the signature for m.
- However, S' is clearly insecure if SHA-1 is used as the hash function.

Canetti-Goldreich-Halevi example

- Suppose signature scheme S is secure in the random oracle model (with random function H).
- Make the following modification to S to obtain a scheme S' that is also secure in the random oracle model:
  - If $H(m) =$ SHA-1(0) then include the private key in the signature for m.
- However, S' is clearly insecure if SHA-1 is used as the hash function.

- The example is extended so that S' is insecure no matter what real-world hash function is used.

General issues with provable security

- Are the definitions the ‘right’ ones?
  - Signatures, public-key encryption, key establishment, etc.
  - No resistance to side-channel attacks.

General issues with provable security

- Are the definitions the ‘right’ ones?
  - Signatures, public-key encryption, key establishment, etc.
  - No resistance to side-channel attacks.
- Are the proofs correct?
  - The security reductions are intricate and lengthy.
  - Halevi (2005): “...as a community, we generate more proofs than we carefully verify...”
  - Halevi (2005): “...I do not expect anyone in his right mind to even read the proof, let alone carefully verify it.”

General issues with provable security

- Are the definitions the ‘right’ ones?
  - Signatures, public-key encryption, key establishment, etc.
  - No resistance to side-channel attacks.
- Are the proofs correct?
  - The security reductions are intricate and lengthy.
  - Halevi (2005): “...as a community, we generate more proofs than we carefully verify...”
  - Halevi (2005): “...I do not expect anyone in his right mind to even read the proof, let alone carefully verify it.”
- What do the proofs really mean?
  - Open problem: Devise a public-key protocol P so that (i) a computational problem X has an optimal but non-tight reduction to the problem of breaking P; (ii) if the parameters for P are selected so that breaking the related problem X is intractable, then P can be broken.

Some questions to ponder

Let A, B be two cryptographic protocols for the same task.

Some questions to ponder

Let A, B be two cryptographic protocols for the same task.

1. If A has a security proof and B doesn't, is A necessarily better than B?
   - Example: DSA versus Schnorr.

Some questions to ponder

Let A, B be two cryptographic protocols for the same task.

1. If A has a security proof and B doesn't, is A necessarily better than B?
   - Example: DSA versus Schnorr.
2. If both A and B have security proofs, which is better?
   - Depends on the assumptions (e.g. Rabin versus RSA)

Some questions to ponder

Let A, B be two cryptographic protocols for the same task.

1. If A has a security proof and B doesn't, is A necessarily better than B?
   - Example: DSA versus Schnorr.
2. If both A and B have security proofs, which is better?
   - Depends on the assumptions (e.g. Rabin versus RSA)
3. If A, B have security proofs under the same assumptions but the reduction for A is tighter, which is better?
   - Example: RSA signatures (FDH vs PSS vs KW).

Some questions to ponder

4. If the reduction for A is highly non-tight, should we have any confidence in the security of A?
   - Example: Schnorr signatures.

Some questions to ponder

4. If the reduction for A is highly non-tight, should we have any confidence in the security of A?
   - Example: Schnorr signatures.
5. If A has a security proof in the random oracle model, but B has a proof in a standard model (but with a non-standard computational assumption), which is better?
   - Example: Identity-based encryption.

Some questions to ponder

4. If the reduction for A is highly non-tight, should we have any confidence in the security of A?
   - Example: Schnorr signatures.
5. If A has a security proof in the random oracle model, but B has a proof in a standard model (but with a non-standard computational assumption), which is better?
   - Example: Identity-based encryption.
6. If A has a tight security proof under standard assumptions, but is twice as slow as competing protocols, should we use A?
   - Example: Cramer-Shoup encryption.

Summary

- Reductionist security arguments are an important tool in the design and analysis of cryptographic protocols.
- More work needs to be done to understand what these reductions really mean.
- Too early to abandon good old-fashioned cryptanalysis and prudent security engineering practices.
- Provable security: Still as much an art as a science.

Further reading

- N. Koblitz and A. Menezes
  Another look at “provable security”
  http://eprint.iacr.org/2004/152.
- N. Koblitz and A. Menezes
  Another look at “provable security”. II
  http://eprint.iacr.org/2006/229.
- N. Koblitz and A. Menezes
  Another look at generic groups
  http://eprint.iacr.org/2006/230.