Another Look at Provable Security

Alfred Menezes

(joint work with Sanjit Chatterjee, Neal Koblitz, Palash Sarkar)

EUROCRYPT 2012

Provable security

Goal: To prove that a protocol $\mathcal{P}$ is secure with respect to a computational problem or primitive $\mathcal{S}$.

Provable security entails:

1. A security definition that captures the capabilities and goals of the adversary.
2. A statement of assumptions about $\mathcal{S}$.
3. A reductionist security proof: $\mathcal{S} \leq \mathcal{A}$, where $\mathcal{A}$ is a hypothetical adversary who breaks $\mathcal{P}$.

Provable security

Goal: To prove that a protocol $\mathcal{P}$ is secure with respect to a computational problem or primitive $\mathcal{S}$.

Provable security entails:

1. A security definition that captures the capabilities and goals of the adversary.
2. A statement of assumptions about $\mathcal{S}$.
3. A reductionist security proof: $\mathcal{S} \leq \mathcal{A}$, where $\mathcal{A}$ is a hypothetical adversary who breaks $\mathcal{P}$.

Question: What security assurances does the proof provide when protocol $\mathcal{P}$ is deployed in practice?

Provable security

Goal: To prove that a protocol $\mathcal{P}$ is secure with respect to a computational problem or primitive $\mathcal{S}$.

Provable security entails:

1. A security definition that captures the capabilities and goals of the adversary.
2. A statement of assumptions about $\mathcal{S}$.
3. A reductionist security proof: $\mathcal{S} \leq \mathcal{A}$, where $\mathcal{A}$ is a hypothetical adversary who breaks $\mathcal{P}$.

Question: What security assurances does the proof provide when protocol $\mathcal{P}$ is deployed in practice?

This talk will examine three difficulties with assessing security proofs: (i) Tightness of the proof; (ii) Multi-user setting; (iii) Non-uniform complexity model.

For concreteness, I will focus on MAC schemes.

What this talk is about

- This talk is about practice-oriented provable security.
  - Understanding what security assurances are provided in practice.

What this talk is about

- This talk is about practice-oriented provable security.
  - Understanding what security assurances are provided in practice.
- This talk is *not* about the foundations of cryptography.

What this talk is about

- This talk is about practice-oriented provable security.
  - Understanding what security assurances are provided in practice.
- This talk is not about the foundations of cryptography.
- This talk is based on papers available at http://anotherlook.ca.
  - These papers are viewed by many as highly controversial.

What this talk is about

- This talk is about practice-oriented provable security.
  - Understanding what security assurances are provided in practice.
- This talk is not about the foundations of cryptography.
- This talk is based on papers available at http://anotherlook.ca.
  - These papers are viewed by many as highly controversial.
  - Anonymous referee: “These papers have elicited a wide variety of reactions from the cryptographic community, ranging from visceral hatred to adulation.”

What this talk is about

- This talk is about practice-oriented provable security.
  - Understanding what security assurances are provided in practice.
- This talk is not about the foundations of cryptography.
- This talk is based on papers available at http://anotherlook.ca.
  - These papers are viewed by many as highly controversial.
  - Anonymous referee: “These papers have elicited a wide variety of reactions from the cryptographic community, ranging from visceral hatred to adulation.”
  - Anonymous referee (in reference to our criticisms of the field of leakage resilience): “What, one must wonder, lies behind this desire to commit infanticide?”

What this talk is about

- This talk is about practice-oriented provable security.
  - Understanding what security assurances are provided in practice.
- This talk is not about the foundations of cryptography.
- This talk is based on papers available at http://anotherlook.ca.
  - These papers are viewed by many as highly controversial.
  - Anonymous referee: “These papers have elicited a wide variety of reactions from the cryptographic community, ranging from visceral hatred to adulation.”
  - Anonymous referee (in reference to our criticisms of the field of leakage resilience): “What, one must wonder, lies behind this desire to commit infanticide?”
- Disclaimer: No babies were killed in preparation for this talk.

Does Tightness Matter?

Tightness gap

- $\mathcal{P}$ = protocol, $\mathcal{S}$ = computational problem/primitive.
- Suppose $\mathcal{A}$ is an algorithm that breaks $\mathcal{P}$. Suppose $\mathcal{A}$ takes time at most T and is successful with probability at least ϵ.
- A reduction of $\mathcal{S}$ to $\mathcal{A}$ (written $\mathcal{S} \leq \mathcal{A}$) is an algorithm $\mathcal{R}$ that solves $\mathcal{S}$ using $\mathcal{A}$ as a subroutine.
- Suppose that $\mathcal{R}$ takes time T' for a proportion at least ϵ' of the instances of $\mathcal{S}$.
- Thus, if $\mathcal{S}$ is (T', ϵ')-secure, then $\mathcal{P}$ is (T, ϵ)-secure.

Tightness gap

- $\mathcal{P}$ = protocol, $\mathcal{S}$ = computational problem/primitive.
- Suppose $\mathcal{A}$ is an algorithm that breaks $\mathcal{P}$. Suppose $\mathcal{A}$ takes time at most T and is successful with probability at least ϵ.
- A reduction of $\mathcal{S}$ to $\mathcal{A}$ (written $\mathcal{S} \leq \mathcal{A}$) is an algorithm $\mathcal{R}$ that solves $\mathcal{S}$ using $\mathcal{A}$ as a subroutine.
- Suppose that $\mathcal{R}$ takes time T' for a proportion at least ϵ' of the instances of $\mathcal{S}$.
- Thus, if $\mathcal{S}$ is (T', ϵ')-secure, then $\mathcal{P}$ is (T, ϵ)-secure.
- The reduction $\mathcal{R}$ is tight if $T' \approx T$ and $\epsilon' \approx \epsilon$.

Tightness gap

- $\mathcal{P}$ = protocol, $\mathcal{S}$ = computational problem/primitive.
- Suppose $\mathcal{A}$ is an algorithm that breaks $\mathcal{P}$. Suppose $\mathcal{A}$ takes time at most T and is successful with probability at least ϵ.
- A reduction of $\mathcal{S}$ to $\mathcal{A}$ (written $\mathcal{S} \leq \mathcal{A}$) is an algorithm $\mathcal{R}$ that solves $\mathcal{S}$ using $\mathcal{A}$ as a subroutine.
- Suppose that $\mathcal{R}$ takes time T' for a proportion at least ϵ' of the instances of $\mathcal{S}$.
- Thus, if $\mathcal{S}$ is (T', ϵ')-secure, then $\mathcal{P}$ is (T, ϵ)-secure.
- The reduction $\mathcal{R}$ is tight if $T' \approx T$ and $\epsilon' \approx \epsilon$.
  It is non-tight if $T \ll T'$ or if $\epsilon \gg \epsilon'$.
- The tightness gap is $(T'\epsilon)/(T\epsilon')$.

Example of a non-tight reduction

The classic Bellare-Rogaway proof for RSA-FDH in the random oracle model has a tightness gap of q, where q is the number of hash function queries.

Example of a non-tight reduction

The classic Bellare-Rogaway proof for RSA-FDH in the random oracle model has a tightness gap of q, where q is the number of hash function queries.

- Let the RSA modulus N be a 1024-bit integer.
- Assumption: The RSA problem cannot be (T', ϵ')-solved for $T'/\epsilon' \leq 2^{80}$.

Example of a non-tight reduction

The classic Bellare-Rogaway proof for RSA-FDH in the random oracle model has a tightness gap of q, where q is the number of hash function queries.

- Let the RSA modulus N be a 1024-bit integer.
- Assumption: The RSA problem cannot be (T', ϵ')-solved for $T'/\epsilon' \leq 2^{80}$.
- Suppose that a (T, ϵ)-forger $\mathcal{A}$ of RSA-FDH makes at most $q = 2^{60}$ hash-queries. Then the Bellare-Rogaway proof uses $\mathcal{A}$ to $(T, \epsilon/2^{60})$-solve the RSA problem.

Example of a non-tight reduction

The classic Bellare-Rogaway proof for RSA-FDH in the random oracle model has a tightness gap of q, where q is the number of hash function queries.

- Let the RSA modulus N be a 1024-bit integer.
- Assumption: The RSA problem cannot be (T', ϵ')-solved for $T'/\epsilon' \leq 2^{80}$.
- Suppose that a (T, ϵ)-forger $\mathcal{A}$ of RSA-FDH makes at most $q = 2^{60}$ hash-queries. Then the Bellare-Rogaway proof uses $\mathcal{A}$ to $(T, \epsilon/2^{60})$-solve the RSA problem.
- Conclusion: RSA-FDH is (T, ϵ)-secure for $T/\epsilon \leq 2^{20}$. The tightness gap is 2^{60}.

Example of a non-tight reduction

The classic Bellare-Rogaway proof for RSA-FDH in the random oracle model has a tightness gap of q, where q is the number of hash function queries.

- Let the RSA modulus N be a 1024-bit integer.
- Assumption: The RSA problem cannot be (T', ϵ')-solved for $T'/\epsilon' \leq 2^{80}$.
- Suppose that a (T, ϵ)-forger $\mathcal{A}$ of RSA-FDH makes at most $q = 2^{60}$ hash-queries. Then the Bellare-Rogaway proof uses $\mathcal{A}$ to $(T, \epsilon/2^{60})$-solve the RSA problem.
- Conclusion: RSA-FDH is (T, ϵ)-secure for $T/\epsilon \leq 2^{20}$. The tightness gap is 2^{60}.
- If we desire the assurance that RSA-FDH is (T, ϵ)-secure for $T/\epsilon \leq 2^{80}$, we need to select N so that $T'/\epsilon' \leq 2^{140}$. That is, we should use at least a 4000-bit modulus N.
- However, no one would take such a recommendation seriously.

Identity-based encryption schemes

- Boyen [2008] compares the tightness of the reductions for the Boneh-Franklin (BF), Sakai-Kasahara (SK), and Boneh-Boyen (BB1) IBE schemes.

Identity-based encryption schemes

- Boyen [2008] compares the tightness of the reductions for the Boneh-Franklin (BF), Sakai-Kasahara (SK), and Boneh-Boyen (BB1) IBE schemes.
- The reduction for BB1 is significantly tighter than the reduction for BF, which in turn is significantly tighter than that for SK.
- However, all three reductions are in fact highly non-tight — the tightness gap being (at least) linear, quadratic and cubic in the number of random oracle queries made by the adversary for BB1, BF and SK, respectively.

Identity-based encryption schemes

- Boyen [2008] compares the tightness of the reductions for the Boneh-Franklin (BF), Sakai-Kasahara (SK), and Boneh-Boyen (BB1) IBE schemes.
- The reduction for BB1 is significantly tighter than the reduction for BF, which in turn is significantly tighter than that for SK.
- However, all three reductions are in fact highly non-tight — the tightness gap being (at least) linear, quadratic and cubic in the number of random oracle queries made by the adversary for BB1, BF and SK, respectively.
- Boyen's recommendations: SK should "generally be avoided as a rule of thumb", BF is "safe to use", and BB1 "appears to be the smartest choice" in part due to the "fairly efficient security reduction" of the latter.

Identity-based encryption schemes

- Boyen [2008] compares the tightness of the reductions for the Boneh-Franklin (BF), Sakai-Kasahara (SK), and Boneh-Boyen (BB1) IBE schemes.
- The reduction for BB1 is significantly tighter than the reduction for BF, which in turn is significantly tighter than that for SK.
- However, all three reductions are in fact highly non-tight — the tightness gap being (at least) linear, quadratic and cubic in the number of random oracle queries made by the adversary for BB1, BF and SK, respectively.
- Boyen's recommendations: SK should "generally be avoided as a rule of thumb", BF is "safe to use", and BB1 "appears to be the smartest choice" in part due to the "fairly efficient security reduction" of the latter.
- However, a recent IETF standard co-authored by Boyen that describes BB1 and BF does not recommend larger security parameters to account for the tightness gaps.

Does tightness matter?

Does tightness matter?

1. Even though the reduction is not tight, it is reasonable to expect that in the future a tighter reduction will be found.

Does tightness matter?

1. Even though the reduction is not tight, it is reasonable to expect that in the future a tighter reduction will be found.
2. Perhaps a tight reduction cannot be found, but a small modification of the protocol can be made in such a way as to permit the construction of a tight reduction.

Does tightness matter?

1. Even though the reduction is not tight, it is reasonable to expect that in the future a tighter reduction will be found.
2. Perhaps a tight reduction cannot be found, but a small modification of the protocol can be made in such a way as to permit the construction of a tight reduction.
3. A tight reduction perhaps can be obtained by relaxing the underlying hard problem.

Does tightness matter?

1. Even though the reduction is not tight, it is reasonable to expect that in the future a tighter reduction will be found.
2. Perhaps a tight reduction cannot be found, but a small modification of the protocol can be made in such a way as to permit the construction of a tight reduction.
3. A tight reduction perhaps can be obtained by relaxing the underlying hard problem.
4. Maybe the notion of security is too strict, and one should relax it a little so as to make possible a tight reduction.

Does tightness matter?

1. Even though the reduction is not tight, it is reasonable to expect that in the future a tighter reduction will be found.
2. Perhaps a tight reduction cannot be found, but a small modification of the protocol can be made in such a way as to permit the construction of a tight reduction.
3. A tight reduction perhaps can be obtained by relaxing the underlying hard problem.
4. Maybe the notion of security is too strict, and one should relax it a little so as to make possible a tight reduction.
5. Perhaps the protocol is secure in practice, even though a tight reduction may simply not exist.

Does tightness matter?

1. Even though the reduction is not tight, it is reasonable to expect that in the future a tighter reduction will be found.
2. Perhaps a tight reduction cannot be found, but a small modification of the protocol can be made in such a way as to permit the construction of a tight reduction.
3. A tight reduction perhaps can be obtained by relaxing the underlying hard problem.
4. Maybe the notion of security is too strict, and one should relax it a little so as to make possible a tight reduction.
5. Perhaps the protocol is secure in practice, even though a tight reduction may simply not exist.
6. Even a non-tight reduction is better than nothing at all.

Does tightness matter?

1. Even though the reduction is not tight, it is reasonable to expect that in the future a tighter reduction will be found.
2. Perhaps a tight reduction cannot be found, but a small modification of the protocol can be made in such a way as to permit the construction of a tight reduction.
3. A tight reduction perhaps can be obtained by relaxing the underlying hard problem.
4. Maybe the notion of security is too strict, and one should relax it a little so as to make possible a tight reduction.
5. Perhaps the protocol is secure in practice, even though a tight reduction may simply not exist.
6. Even a non-tight reduction is better than nothing at all.
7. [nightmare scenario] Perhaps the protocol is in fact insecure, but an attack has not yet been discovered.

MACs in the multi-user setting

- Let $H_k : \{0,1\}^* \rightarrow \{0,1\}^t$ be a family of MAC functions, where $k \in \{0,1\}^r$.
- Let $k \in_R \{0,1\}^r$ be the secret key. The standard security definition for MAC schemes is that an adversary $\mathcal{B}$ who has access to an oracle for H_k is unable to produce a valid message-tag pair (where the message was not queried to the oracle). Call the adversary's task breaking MAC1.

MACs in the multi-user setting

- Let $H_k : \{0,1\}^* \to \{0,1\}^t$ be a family of MAC functions, where $k \in \{0,1\}^r$.

- Let $k \in_R \{0,1\}^r$ be the secret key. The standard security definition for MAC schemes is that an adversary $\mathcal{B}$ who has access to an oracle for H_k is unable to produce a valid message-tag pair (where the message was not queried to the oracle). Call the adversary's task breaking MAC1.

- Consider using the same MAC scheme in a multi-user setting. Let $k_1, k_2, \ldots, k_n \in_R \{0,1\}^r$. The adversary $\mathcal{A}$ has access to oracles for H_{k_i}. Her task is to produce a triple (i, m, τ), where $1 \leq i \leq n$, $H_{k_i}(m) = \tau$, and m was not queried to H_{k_i}. Call the adversary's task breaking MAC*.

Security proof for MAC*

- The proof is a reduction from breaking MAC1 to breaking MAC*.

Security proof for MAC*

- The proof is a reduction from breaking MAC1 to breaking MAC*.
- Let $\mathcal{A}$ be an adversary that (T, ϵ)-breaks MAC*. We are given an oracle for H_k, where $k \in_R \{0,1\}^r$. Our goal is to produce a MAC forgery with respect to H_k.

Security proof for MAC*

- The proof is a reduction from breaking MAC1 to breaking MAC*.
- Let $\mathcal{A}$ be an adversary that (T, ϵ)-breaks MAC*. We are given an oracle for H_k, where $k \in_R \{0,1\}^r$. Our goal is to produce a MAC forgery with respect to H_k.
- Select $j \in_R [1, n]$.
- For each $i \in [1, n]$ with $i \neq j$, select $k_i \in_R \{0,1\}^r$ as i's secret key. User j's secret key is assigned to be k.
- Run $\mathcal{A}$, using k_i's to answer $\mathcal{A}$'s MAC queries to users $i \neq j$, and the given oracle H_k to answer $\mathcal{A}$'s MAC queries to user j.
- If $\mathcal{A}$ outputs a forgery (j, m, τ), then output (m, τ) as a forgery with respect to H_k.
- Success probability is ϵ/n.

Security proof for MAC*

- The proof is a reduction from breaking MAC1 to breaking MAC*.
- Let $\mathcal{A}$ be an adversary that (T, ϵ)-breaks MAC*. We are given an oracle for H_k, where $k \in_R \{0,1\}^r$. Our goal is to produce a MAC forgery with respect to H_k.
- Select $j \in_R [1, n]$.
- For each $i \in [1, n]$ with $i \neq j$, select $k_i \in_R \{0,1\}^r$ as i's secret key. User j's secret key is assigned to be k.
- Run $\mathcal{A}$, using k_i's to answer $\mathcal{A}$'s MAC queries to users $i \neq j$, and the given oracle H_k to answer $\mathcal{A}$'s MAC queries to user j.
- If $\mathcal{A}$ outputs a forgery (j, m, τ), then output (m, τ) as a forgery with respect to H_k.
- Success probability is ϵ/n.
- Summary: if MAC1 is (T', ϵ')-secure, then MAC* is $(T', n\epsilon')$-secure. The tightness gap is n.

Provably secure, but insecure

An attack on MAC*: [Biham's key collision attack]

- Suppose that $r \leq t$.
- Select a single arbitrary m and obtain tags $H_{k_i}(m)\ \forall i \in [1, n]$.
- Select an arbitrary subset $\mathcal{W}$ of keys with $|\mathcal{W}| = w$.
- For each $\ell \in \mathcal{W}$, compute $H_\ell(m)$; if $H_\ell(m) = H_{k_i}(m)$ for some i, then conclude that $\ell = k_i$ and use k_i to forge a message-tag pair for i.

Provably secure, but insecure

An attack on MAC*: [Biham's key collision attack]

- Suppose that $r \leq t$.
- Select a single arbitrary m and obtain tags $H_{k_i}(m)\ \forall i \in [1, n]$.
- Select an arbitrary subset $\mathcal{W}$ of keys with $|\mathcal{W}| = w$.
- For each $\ell \in \mathcal{W}$, compute $H_\ell(m)$; if $H_\ell(m) = H_{k_i}(m)$ for some i, then conclude that $\ell = k_i$ and use k_i to forge a message-tag pair for i.

- Example: CMAC with 80-bit keys and 80-bit tags. Assume that $n = 2^{20}$. Choose $w = 2^{60}$, so that the attack takes time 2^{60}. Time-memory trade-off: With an offline computation of 2^{60} MAC computations, and 2^{40} storage units, the adversary can find one of 2^{20} keys with an on-line search time of 2^{40}.

Provably secure, but insecure

An attack on MAC*: [Biham's key collision attack]

- Suppose that $r \leq t$.
- Select a single arbitrary m and obtain tags $H_{k_i}(m)\ \forall i \in [1,n]$.
- Select an arbitrary subset $\mathcal{W}$ of keys with $|\mathcal{W}| = w$.
- For each $\ell \in \mathcal{W}$, compute $H_\ell(m)$; if $H_\ell(m) = H_{k_i}(m)$ for some i, then conclude that $\ell = k_i$ and use k_i to forge a message-tag pair for i.

- Example: CMAC with 80-bit keys and 80-bit tags. Assume that $n = 2^{20}$. Choose $w = 2^{60}$, so that the attack takes time 2^{60}. Time-memory trade-off: With an offline computation of 2^{60} MAC computations, and 2^{40} storage units, the adversary can find one of 2^{20} keys with an on-line search time of 2^{40}.
- Note: Speedup over the generic key-finding attack on MAC1 is by n. This is the nightmare scenario since the tightness gap translated to an actual practical attack.

A fix: fMAC

- One countermeasure is fMAC: The tag of m is (f, τ) where f is a fixed, non-secret, and unique string shared between the two communicating parties for that session and $\tau = H_k(f, m)$.

A fix: fMAC

- One countermeasure is fMAC: The tag of m is (f, τ) where f is a fixed, non-secret, and unique string shared between the two communicating parties for that session and $\tau = H_k(f, m)$.
- fMAC* resists the previous attack. And MAC* $\leq$ fMAC*.

A fix: fMAC

- One countermeasure is fMAC: The tag of m is (f, τ) where f is a fixed, non-secret, and unique string shared between the two communicating parties for that session and $\tau = H_k(f, m)$.
- fMAC* resists the previous attack. And MAC* $\leq$ fMAC*.
- fMAC* can be proven secure under the assumption that MAC1 is secure. As with MAC*, the tightness gap is n.

A fix: fMAC

- One countermeasure is fMAC: The tag of m is (f, τ) where f is a fixed, non-secret, and unique string shared between the two communicating parties for that session and $\tau = H_k(f, m)$.
- fMAC* resists the previous attack. And MAC* $\leq$ fMAC*.
- fMAC* can be proven secure under the assumption that MAC1 is secure. As with MAC*, the tightness gap is n.
- However, one would expect that fMAC* and MAC1 are tightly related in practice.

A fix: fMAC

- One countermeasure is fMAC: The tag of m is (f, τ) where f is a fixed, non-secret, and unique string shared between the two communicating parties for that session and $\tau{=}H_k(f, m)$.
- fMAC* resists the previous attack. And MAC* $\leq$ fMAC*.
- fMAC* can be proven secure under the assumption that MAC1 is secure. As with MAC*, the tightness gap is n.
- However, one would expect that fMAC* and MAC1 are tightly related in practice.
- From a provable security standpoint, there is little difference between MAC* and fMAC*.

MAC* in other protocols

The tightness gap of the MAC* reduction appears in the security proofs of several protocols including:

- Katz-Lindell aggregate MAC scheme [2008]
- Eikemeier et al. history-free aggregate MAC scheme [2010]
- Canetti-Krawczyk network authentication protocol [2001]

These protocols succumb to attacks like the one on MAC*.

MAC* in other protocols

The tightness gap of the MAC* reduction appears in the security proofs of several protocols including:

- Katz-Lindell aggregate MAC scheme [2008]
- Eikemeier et al. history-free aggregate MAC scheme [2010]
- Canetti-Krawczyk network authentication protocol [2001]

These protocols succumb to attacks like the one on MAC*.

Non-tight security proofs can give one a false sense of security.

MAC* in other protocols

The tightness gap of the MAC* reduction appears in the security proofs of several protocols including:

- Katz-Lindell aggregate MAC scheme [2008]
- Eikemeier et al. history-free aggregate MAC scheme [2010]
- Canetti-Krawczyk network authentication protocol [2001]

These protocols succumb to attacks like the one on MAC*.

Non-tight security proofs can give one a false sense of security.

Question: Are security proofs with non-tight reductions of any practical value?

The Multi-User Setting

Single-user vs. multi-user

- Previous work in the multi-user setting:
  - key establishment, public-key encryption, signatures.

Single-user vs. multi-user

- Previous work in the multi-user setting:
  - key establishment, public-key encryption, signatures.
- The (single-user setting) security definition for MAC schemes is inadequate.
  - One might argue that a secure MAC scheme is a primitive and not a protocol.
  - However, a secure MAC scheme ought to be secure for its primary application — authentication of messages.

Single-user vs. multi-user

- Previous work in the multi-user setting:
  - key establishment, public-key encryption, signatures.
- The (single-user setting) security definition for MAC schemes is inadequate.
  - One might argue that a secure MAC scheme is a primitive and not a protocol.
  - However, a secure MAC scheme ought to be secure for its primary application — authentication of messages.
- Similarly, the GMR security definition for signature schemes is inadequate for the multi-user setting.

Single-user vs. multi-user

- Previous work in the multi-user setting:
  - key establishment, public-key encryption, signatures.
- The (single-user setting) security definition for MAC schemes is inadequate.
  - One might argue that a secure MAC scheme is a primitive and not a protocol.
  - However, a secure MAC scheme ought to be secure for its primary application — authentication of messages.
- Similarly, the GMR security definition for signature schemes is inadequate for the multi-user setting.
- The BGLS security definition for aggregate signature schemes is in the multi-user setting, but is deficient.
  - The adversary is not allowed to adaptively select its target user.

Single-user vs. multi-user

The following schemes succumb to attacks that are analogous to the one on MAC*:

- Rogaway-Shrimpton deterministic authenticated encryption (SIV) [2006]
- OCB authenticated encryption [2003]
- EME disk encryption [2004]
- (Zaverucha 2012) Hybrid encryption (KEM/DEM schemes where the DEM is deterministic).
- (Zaverucha 2012) Krawczyk's extract-then-expand key derivation [2010], as standardized in NIST SP 800-56C and RFC 5869.

Single-user vs. multi-user

The following schemes succumb to attacks that are analogous to the one on MAC*:

- Rogaway-Shrimpton deterministic authenticated encryption (SIV) [2006]
- OCB authenticated encryption [2003]
- EME disk encryption [2004]
- (Zaverucha 2012) Hybrid encryption (KEM/DEM schemes where the DEM is deterministic).
- (Zaverucha 2012) Krawczyk's extract-then-expand key derivation [2010], as standardized in NIST SP 800-56C and RFC 5869.

Question: Should one be suspicious of security definitions and theorems that are in the single-user setting?

The Non-Uniform Complexity Model

HMAC

- For concreteness, we will consider HMAC-MD5.
- Let $f : \{0,1\}^{128} \times \{0,1\}^{512} \longrightarrow \{0,1\}^{128}$ denote the MD5 compression function.
- Let $H_{IV} : \{0,1\}^* \longrightarrow \{0,1\}^{128}$ denote the MD5 iterated hash function with initialization vector IV.
- Then NMAC$_{k_1,k_2}(m) = f(k_1, H_{k_2}(m)^0)$.
- HMAC is a one-key variant of NMAC.

HMAC

- For concreteness, we will consider HMAC-MD5.
- Let $f : \{0,1\}^{128} \times \{0,1\}^{512} \longrightarrow \{0,1\}^{128}$ denote the MD5 compression function.
- Let $H_{IV} : \{0,1\}^{*} \longrightarrow \{0,1\}^{128}$ denote the MD5 iterated hash function with initialization vector IV.
- Then NMAC$_{k_1,k_2}(m) = f(k_1, H_{k_2}(m)^0)$.
- HMAC is a one-key variant of NMAC.
- Bellare-Canetti-Krawczyk's (1996) security proof for NMAC (as a MAC scheme) assumed (i) f is a secure MAC scheme; and (ii) H is collision resistant.
- Wang's (2005) collision finding algorithm for MD5 rendered the proof useless as a security guarantee for NMAC-MD5.

HMAC

- For concreteness, we will consider HMAC-MD5.
- Let $f : \{0,1\}^{128} \times \{0,1\}^{512} \longrightarrow \{0,1\}^{128}$ denote the MD5 compression function.
- Let $H_{IV} : \{0,1\}^* \longrightarrow \{0,1\}^{128}$ denote the MD5 iterated hash function with initialization vector IV.
- Then NMAC$_{k_1,k_2}(m) = f(k_1, H_{k_2}(m)^0)$.
- HMAC is a one-key variant of NMAC.
- Bellare-Canetti-Krawczyk's (1996) security proof for NMAC (as a MAC scheme) assumed (i) f is a secure MAC scheme; and (ii) H is collision resistant.
- Wang's (2005) collision finding algorithm for MD5 rendered the proof useless as a security guarantee for NMAC-MD5.
- Bellare (2006) gave a new proof for the security for NMAC as a pseudorandom function (prf). The proof only assumed that f is a secure prf.

Bellare's security theorem for NMAC

- Theorem: If f is a secure prf, then NMAC is a secure prf.

Bellare's security theorem for NMAC

- Theorem: If f is a secure prf, then NMAC is a secure prf.

- The proof is in the non-uniform complexity model.
- In this model, an "algorithm" is a sequence of Boolean circuits, one for each input size. One is only concerned with the existence of these Boolean circuits, regardless of whether there is a feasible way to construct the circuits.
- Another way to think of a non-uniform algorithm is as a Turing machine with (polynomial-size) advice strings which depend on the input length but not on the input itself. These advice strings need only exist in the mathematical sense and not be constructible in any practical sense.

Bellare's security theorem for NMAC

- Theorem: If f is a secure prf, then NMAC is a secure prf.
- Security proofs in the non-uniform complexity model have been claimed by some to be desirable because their conclusions are stronger than in the uniform model.
  - (Goldwasser, 1990) "The most meaningful proofs of security are necessarily those proved with respect to the most powerful adversary. To this end, we should let the polynomial-time adversary be not only probabilistic but also nonuniform."

Bellare's security theorem for NMAC

- Theorem: If f is a secure prf, then NMAC is a secure prf.
- Security proofs in the non-uniform complexity model have been claimed by some to be desirable because their conclusions are stronger than in the uniform model.
  - (Goldwasser, 1990) "The most meaningful proofs of security are necessarily those proved with respect to the most powerful adversary. To this end, we should let the polynomial-time adversary be not only probabilistic but also nonuniform."
- In fact, they are less desirable because it is extremely difficult to assess the difficulty of the hypotheses in the non-uniform model. Also, the hypotheses are typically stronger in the non-uniform model than they would be in the uniform model. [see the Bernstein/Lange Eurocrypt 2012 rump session talk].

PRF security

- Security assumption: f is (t, ϵ, q)-secure. That is, adversaries with running time $\leq t$, and making $\leq q$ oracle queries, have advantage $\leq \epsilon$ of deciding whether a given oracle O is a random function or f with hidden key.

PRF security

- Security assumption: f is (t, ϵ, q)-secure. That is, adversaries with running time $\leq t$, and making $\leq q$ oracle queries, have advantage $\leq \epsilon$ of deciding whether a given oracle O is a random function or f with hidden key.
- For MD5, the fastest known algorithm in the uniform model for breaking prfness is exhaustive key search: in the course of its running time t, the adversary is able to try t keys and so its advantage is $\epsilon = t/2^{128}$ (so $t/\epsilon = 2^{128}$).

PRF security

- Security assumption: f is (t, ϵ, q)-secure. That is, adversaries with running time $\leq t$, and making $\leq q$ oracle queries, have advantage $\leq \epsilon$ of deciding whether a given oracle O is a random function or f with hidden key.
- For MD5, the fastest known algorithm in the uniform model for breaking prfness is exhaustive key search: in the course of its running time t, the adversary is able to try t keys and so its advantage is $\epsilon = t/2^{128}$ (so $t/\epsilon = 2^{128}$).
- When evaluating the conditions under which his theorem has content, Bellare assumes that exhaustive key search is the fastest generic attack for breaking prfness of f.
- However, there are more effective generic algorithms in the non-uniform model.

PRF security in the non-uniform model

- Assume that f has "good randomness properties".
- For $x \in \{0,1\}^{128}$, let $u(x)$ denote a fixed bit of x.
- For each $M \in \{0,1\}^{512}$, let Prob(M) be the probability that $u(f(k,M)) = 1$.
- Let M^* be a message for which Prob(M) is maximum.
- Claim: $\text{Prob}(M^*) > \frac{1}{2} + \frac{1}{2^{64}}$.
  Justification: Fix M. Consider $u(f(k,M))$ as defining a random walk [forward step if $u(f(k,M)) = 1$, backward step if $u(f(k,M)) = 0$]. The standard deviation from the starting point in a random walk with 2^{128} steps is 2^{64}.

PRF security in the non-uniform model

- Assume that f has "good randomness properties".
- For $x \in \{0,1\}^{128}$, let $u(x)$ denote a fixed bit of x.
- For each $M \in \{0,1\}^{512}$, let Prob(M) be the probability that $u(f(k,M)) = 1$.
- Let M^* be a message for which Prob(M) is maximum.
- Claim: Prob$(M^*) > \frac{1}{2} + \frac{1}{2^{64}}$.
  Justification: Fix M. Consider $u(f(k,M))$ as defining a random walk [forward step if $u(f(k,M)) = 1$, backward step if $u(f(k,M)) = 0$]. The standard deviation from the starting point in a random walk with 2^{128} steps is 2^{64}.
- Algorithm for breaking prfness of f: Query M^* to the oracle O. If $u(O(M^*)) = 1$ then guess that the oracle is $f(k,\cdot)$; otherwise guess that the oracle is random.
  Running time $= 1$. Advantage $> \frac{1}{2^{64}}$. # of queries $= 1$.

PRF security in the non-uniform model

- Assume that f has "good randomness properties".
- For $x \in \{0,1\}^{128}$, let $u(x)$ denote a fixed bit of x.
- For each $M \in \{0,1\}^{512}$, let Prob(M) be the probability that $u(f(k,M)) = 1$.
- Let M^* be a message for which Prob(M) is maximum.
- Claim: Prob$(M^*) > \frac{1}{2} + \frac{1}{2^{64}}$.
  Justification: Fix M. Consider $u(f(k,M))$ as defining a random walk [forward step if $u(f(k,M)) = 1$, backward step if $u(f(k,M)) = 0$]. The standard deviation from the starting point in a random walk with 2^{128} steps is 2^{64}.
- Algorithm for breaking prfness of f: Query M^* to the oracle O. If $u(O(M^*)) = 1$ then guess that the oracle is $f(k,\cdot)$; otherwise guess that the oracle is random.
  Running time $= 1$. Advantage $> \frac{1}{2^{64}}$. # of queries $= 1$.
  The t/ϵ ratio is 2^{64} versus 2^{128} for exhaustive key search.

Interpreting Bellare's proof in practice

- Suppose that messages are $n = 2^{20}$ blocks in length.
- Under the assumption that exhaustive key search is the fastest attack on the prfness of f, Bellare argues that his proof justifies NMAC-MD5 up to 2^{44} queries (and 2^{60} queries for NMAC-SHA1).

Interpreting Bellare's proof in practice

- Suppose that messages are $n = 2^{20}$ blocks in length.
- Under the assumption that exhaustive key search is the fastest attack on the prfness of f, Bellare argues that his proof justifies NMAC-MD5 up to 2^{44} queries (and 2^{60} queries for NMAC-SHA1).
- However, Bellare's proof uses coin-fixing: When converting a prf-adversary of NMAC to a prf-adversary of f, the run time of the prf-adversary of f is significantly reduced (thus yielding a significant improvement in the tightness of the reduction). This adversary is unconstructible.

Interpreting Bellare's proof in practice

- Suppose that messages are $n = 2^{20}$ blocks in length.
- Under the assumption that exhaustive key search is the fastest attack on the prfness of f, Bellare argues that his proof justifies NMAC-MD5 up to 2^{44} queries (and 2^{60} queries for NMAC-SHA1).
- However, Bellare's proof uses coin-fixing: When converting a prf-adversary of NMAC to a prf-adversary of f, the run time of the prf-adversary of f is significantly reduced (thus yielding a significant improvement in the tightness of the reduction). This adversary is unconstructible.
- In light of the faster prf-adversary in the non-uniform model that was described above, Bellare's proof says nothing about NMAC-MD5 security if $q > 2^{22}$ queries.
- Similarly, Bellare's proof says nothing about NMAC-SHA1 security if $q > 2^{30}$.

Is HMAC-MD5 provably secure?

Is HMAC-MD5 provably secure?

- In [KM 2012], we gave a tighter proof, in the uniform model, that NMAC is a secure prf assuming only that f is a secure prf.
- The proof justifies NMAC-MD5 up to 2^{54} queries, and NMAC-SHA1 up to 2^{70} queries.

Is HMAC-MD5 provably secure?

- In [KM 2012], we gave a tighter proof, in the uniform model, that NMAC is a secure prf assuming only that f is a secure prf.
- The proof justifies NMAC-MD5 up to 2^{54} queries, and NMAC-SHA1 up to 2^{70} queries.
- However:
  - The proof has tightness gap of $9n^2$.
  - The proof is in the single-user setting.
  - Assuming prf-ness of f is still a strong assumption.

  So, the value of our proof as a source of assurance about the real-world security of HMAC-MD5 is questionable at best.

Is HMAC-MD5 provably secure?

- In [KM 2012], we gave a tighter proof, in the uniform model, that NMAC is a secure prf assuming only that f is a secure prf.
- The proof justifies NMAC-MD5 up to 2^{54} queries, and NMAC-SHA1 up to 2^{70} queries.
- However:
  - The proof has tightness gap of $9n^2$.
  - The proof is in the single-user setting.
  - Assuming prf-ness of f is still a strong assumption.

  So, the value of our proof as a source of assurance about the real-world security of HMAC-MD5 is questionable at best.
- Note: Bernstein observed in 2005 that NMAC has a straightforward security proof under the assumptions that (i) f is a secure prf, and (ii) H is an almost-universal hash function.
  - Question: Are MD5 & SHA1 almost-universal hash fns.?

Non-uniform complexity model

Other questionable uses of the non-uniform complexity model in security proofs include:

- Multi-property-preserving hash domain extension (Bellare & Ristenpart, 2006).
- Sandwich hash MAC scheme (Yasuda, 2007).
- Boosting Merkle-Damgård hashing for MACs (Yasuda, 2007).
- Leakage-resilient stream ciphers from pseudorandom bit generators (Dziembowski & Pietrzak, 2008).
- ???

Non-uniform complexity model

Other questionable uses of the non-uniform complexity model in security proofs include:

- Multi-property-preserving hash domain extension (Bellare & Ristenpart, 2006).
- Sandwich hash MAC scheme (Yasuda, 2007).
- Boosting Merkle-Damgård hashing for MACs (Yasuda, 2007).
- Leakage-resilient stream ciphers from pseudorandom bit generators (Dziembowski & Pietrzak, 2008).
- ???

Question: Should unconstructible security proofs in the non-uniform model be rejected?

Concluding remarks

Significance of our work

Significance of our work

- Theoreticians who work in the foundations of cryptography and are not interested in the practicality of theoretical work can safely ignore our results.

Significance of our work

- Theoreticians who work in the foundations of cryptography and are not interested in the practicality of theoretical work can safely ignore our results.

- Practitioners who use security proofs only as one possible tool to assess the security of a cryptographic system, but rely more heavily on extensive cryptanalysis and sound engineering principles, should not be alarmed by our observations.

Significance of our work

- Theoreticians who work in the foundations of cryptography and are not interested in the practicality of theoretical work can safely ignore our results.
- Practitioners who use security proofs only as one possible tool to assess the security of a cryptographic system, but rely more heavily on extensive cryptanalysis and sound engineering principles, should not be alarmed by our observations.
- Cryptographers who believe that a security proof is the essential, and perhaps the only, way to gain confidence in the practical security of a protocol should be much more concerned. They should be skeptical of non-tight proofs, proofs in the single-user setting, and proofs in the non-uniform complexity model, and perhaps even reject these proofs as mere heuristic arguments for the protocol's security.

COPS: Cryptanalysis Of Provable Security

- A lot more work remains to be done to fully understand what practical assurances are provided by the many existing security theorems.

COPS: Cryptanalysis Of Provable Security

- A lot more work remains to be done to fully understand what practical assurances are provided by the many existing security theorems.

- Some important questions that remain unanswered are:
  1. Is a non-tight security proof of any value in practice?
  2. Should one be suspicious of security definitions that are in the single-user setting?
  3. Should unconstructible security proofs in the non-uniform model be rejected?
  4. Are HMAC-MD5 and HMAC-SHA1 provably secure?

- These questions are more relevant to practice than concerns about the random oracle assumption in proofs.

COPS: Cryptanalysis Of Provable Security

- The main goal of practice-oriented cryptographic research should be concrete security assurances, not just mathematical formalism and correctness.

COPS: Cryptanalysis Of Provable Security

- The main goal of practice-oriented cryptographic research should be concrete security assurances, not just mathematical formalism and correctness.

- In connection with the error in the original proof for OAEP, Stern, Pointcheval, Malone-Lee and Smart (2002) comment:
  - “The use of provable security is more subtle than it appears, and flaws in security proofs themselves might have a devastating effect on the trustworthiness of cryptography. By flaws, we do not mean plain mathematical errors but rather ambiguities or misconceptions in the security model.”

A radical proposal

A radical proposal

- An avenue for positive change is to ensure that security proofs start to get the detailed peer review they need:
  - Proofs should not be in the appendices of submitted papers – referees must be required to read the proofs.
  - Full papers should be published, not “extended abstracts”.
  - There shouldn’t be any page limits on published papers.

A radical proposal

- An avenue for positive change is to ensure that security proofs start to get the detailed peer review they need:
  - Proofs should not be in the appendices of submitted papers – referees must be required to read the proofs.
  - Full papers should be published, not “extended abstracts”.
  - There shouldn’t be any page limits on published papers.

- Strive for a better balance of the programs of major crypto conferences:
  - Consider merging PKC/CHES/FSE with Crypto/Eurocrypt/Asiacrypt.
  - Consider allowing parallel sessions.

In conclusion....

While mathematical proofs have their place in cryptography, our work illustrates some limitations of such proofs and highlights the important role that old-fashioned cryptanalysis and sound engineering practices continue to play in establishing and maintaining confidence in the security of a cryptographic system.

http://anotherlook.ca