Better proofs for rekeying

D. J. Bernstein

Security of AES-256 key k is far below 2^{256} in most protocols: $(\text{AES}_k(0), \ldots, \text{AES}_k(n-1))$ is distinguishable from uniform with probability $n(n-1)/2^{129}$, plus tiny key-guessing probability. Yes, distinguishers matter.

Attacker actually has T targets: independent keys $k_1, \ldots, k_T$. Success chance $\approx Tn(n-1)/2^{129}$.

"Rekeying" seems less dangerous.

Expand k into $F(k) =$
$(\text{AES}_k(0), \ldots, \text{AES}_k(999999))$.

Split $F(k)$ into 500000 "subkeys".

Output $F(k')$ for each subkey k':
i.e., $F(\text{AES}_k(0), \text{AES}_k(1))$;
$F(\text{AES}_k(2), \text{AES}_k(3))$; ...
$F(\text{AES}_k(999998), \text{AES}_k(999999))$.

"Rekeying" seems less dangerous.

Expand k into $F(k) =$ $(\text{AES}_k(0), \ldots, \text{AES}_k(999999))$.

Split $F(k)$ into 500000 "subkeys".

Output $F(k')$ for each subkey k': i.e., $F(\text{AES}_k(0), \text{AES}_k(1))$; $F(\text{AES}_k(2), \text{AES}_k(3))$; ... $F(\text{AES}_k(999998), \text{AES}_k(999999))$.

Repeat for $k_1, \ldots, k_T$. **What is attacker's success chance p_T?**

"Rekeying" seems less dangerous.

Expand k into $F(k) =$ $(\text{AES}_k(0), \ldots, \text{AES}_k(999999))$.

Split $F(k)$ into 500000 "subkeys".

Output $F(k')$ for each subkey k': i.e., $F(\text{AES}_k(0), \text{AES}_k(1))$; $F(\text{AES}_k(2), \text{AES}_k(3))$; ... $F(\text{AES}_k(999998), \text{AES}_k(999999))$.

Repeat for $k_1, \ldots, k_T$. **What is attacker's success chance p_T?**

Intuitively clear that $p_T \le T p_1$. So let's analyze p_1.

Attack strategy 1: Attack the master key k. Distinguish $F(k)$ from a uniform random string.

Years of cryptanalysis say: hard to distinguish AES outputs from uniform string of distinct blocks. Distinctness loses $\approx 1/2^{89}$.

Attack strategy 1: Attack the master key k. Distinguish $F(k)$ from a uniform random string.

Years of cryptanalysis say: hard to distinguish AES outputs from uniform string of distinct blocks. Distinctness loses $\approx 1/2^{89}$.

Attack strategy 2: Attack a subkey k'. Distinguish $F(k')$ from uniform, assuming k' is uniform.

Attack strategy 1: Attack the master key k. Distinguish $F(k)$ from a uniform random string.

Years of cryptanalysis say: hard to distinguish AES outputs from uniform string of distinct blocks. Distinctness loses $\approx 1/2^{89}$.

Attack strategy 2: Attack a subkey k'. Distinguish $F(k')$ from uniform, assuming k' is uniform.

Intuition: No other attacks exist.
But where is this proven?

FOCS 1996 Bellare–Canetti–Krawczyk claims to prove security of ℓ-level "cascade".

2-level cascade: key k; input (N_1, N_2); output $S(S(k, N_1), N_2)$.

FOCS 1996 Bellare–Canetti–Krawczyk claims to prove security of ℓ-level "cascade".

2-level cascade: key k; input (N_1, N_2); output $S(S(k, N_1), N_2)$.

Example: Define $S(k, N) = (\text{AES}_k(2N), \text{AES}_k(2N + 1))$, with $N \in \{0, 1, \ldots, 499999\}$.
S expands AES-256 key k into $(\text{AES}_k(0), \ldots, \text{AES}_k(999999))$.

FOCS 1996 Bellare–Canetti–Krawczyk claims to prove security of ℓ-level "cascade".

2-level cascade: key k; input (N_1, N_2); output $S(S(k, N_1), N_2)$.

Example: Define $S(k, N) = (\text{AES}_k(2N), \text{AES}_k(2N+1))$, with $N \in \{0, 1, \ldots, 499999\}$.
S expands AES-256 key k into $(\text{AES}_k(0), \ldots, \text{AES}_k(999999))$.

Paper credits 1986 Goldwasser–Goldreich–Micali for 1-bit N_i:
S expands k into $S(k, 0), S(k, 1)$.

Theorem statement is wrong:
omits factor q. Fixed in 2005.

Here q is the number of queries.
The intuition didn't notice q;
why does q matter for the proof?

Theorem statement is wrong: omits factor q. Fixed in 2005.

Here q is the number of queries. The intuition didn't notice q; why does q matter for the proof?

Proof outline: Take any cascade attack A using at most q queries.

Proof has $q + 1$ steps.

Theorem statement is wrong: omits factor q. Fixed in 2005.

Here q is the number of queries. The intuition didn't notice q; why does q matter for the proof?

Proof outline: Take any cascade attack A using at most q queries.

Proof has $q+1$ steps.

Step 0: Replace outputs from master key k with independent uniform random outputs.

Distinguisher for this step $\Rightarrow$ attack against S.

Step 1: Replace cascade outputs for *first* subkey with independent uniform random outputs.

Distinguisher for this step
$\Rightarrow$ attack against S.

Step 1: Replace cascade outputs for *first* subkey with independent uniform random outputs.

Distinguisher for this step
$\Rightarrow$ attack against S.

Step 2: Replace cascade outputs from next (distinct) subkey. . . .
Step q: Replace cascade outputs from qth (distinct) subkey.
Could skip steps if $q > \#\{N\}$.

Step 1: Replace cascade outputs for *first* subkey with independent uniform random outputs.

Distinguisher for this step $\Rightarrow$ attack against S.

Step 2: Replace cascade outputs from next (distinct) subkey. ...
Step q: Replace cascade outputs from qth (distinct) subkey.
Could skip steps if $q > \#\{N\}$.

Further complications in proof to monolithically handle ℓ levels.
2011 Bernstein: simpler to compose better 2-level theorem.

Not happy with cascade proofs?

A different proof appears in Crypto 1996 Bellare–Canetti–Krawczyk NMAC/HMAC paper.

Not happy with cascade proofs?

A different proof appears in Crypto 1996 Bellare–Canetti–Krawczyk NMAC/HMAC paper.

Given key k and input (N_1, N_2), NMAC computes $S(S(k, N_1), N_2)$, where S is a ~~stream cipher~~ "compression function".

(Tweaks: output is encrypted; no prefix-free requirement.)

Not happy with cascade proofs?

A different proof appears in Crypto 1996 Bellare–Canetti–Krawczyk NMAC/HMAC paper.

Given key k and input (N_1, N_2), NMAC computes $S(S(k, N_1), N_2)$, where S is a ~~stream cipher~~ "compression function".

(Tweaks: output is encrypted; no prefix-free requirement.)

Proof has weird assumptions. Crypto 2006 Bellare proof: more reasonable-sounding assumptions.

Complicated; error-prone.

2012 Koblitz–Menezes:
Bellare's assumptions are wrong.

Complicated; error-prone.

2012 Koblitz–Menezes:
Bellare's assumptions are wrong.

2012 Katz–Lindell: public denials.

Complicated; error-prone.

2012 Koblitz–Menezes:
Bellare's assumptions are wrong.

2012 Katz–Lindell: public denials.

2012 Bernstein–Lange:
Bellare's assumptions are wrong.

Complicated; error-prone.

2012 Koblitz–Menezes:
Bellare's assumptions are wrong.

2012 Katz–Lindell: public denials.

2012 Bernstein–Lange:
Bellare's assumptions are wrong.

2013 Pietrzak: fixed theorem
from Koblitz–Menezes is wrong.

Complicated; error-prone.

2012 Koblitz–Menezes:
Bellare's assumptions are wrong.

2012 Katz–Lindell: public denials.

2012 Bernstein–Lange:
Bellare's assumptions are wrong.

2013 Pietrzak: fixed theorem
from Koblitz–Menezes is wrong.

2013 Pietrzak, 2013 Koblitz–
Menezes, 2014 Gaži–Pietrzak–
Rybár: another NMAC proof,
as complicated as cascade proof.

Hmmm. CCS 2005 Barak–Halevi "A model and architecture for pseudo-random generation with applications to `/dev/random`"?

RNG outputs $F(k)$, $F(G(k))$, etc.

Another complicated proof.

Hmmm. CCS 2005 Barak–Halevi "A model and architecture for pseudo-random generation with applications to `/dev/random`"? RNG outputs $F(k)$, $F(G(k))$, etc. Another complicated proof.

How about 2006 Campagna "Security bounds for the NIST codebook-based deterministic random bit generator"? Doesn't prove anything about rekeying.

Hmmm. CCS 2005 Barak–Halevi "A model and architecture for pseudo-random generation with applications to `/dev/random`"? RNG outputs $F(k)$, $F(G(k))$, etc. Another complicated proof.

How about 2006 Campagna "Security bounds for the NIST codebook-based deterministic random bit generator"? Doesn't prove anything about rekeying.

2017 AES-GCM-SIV bounds? Big errors found by Iwata–Seurin.

A simple tight new proof

Remember the goal: analyze p_T.

There are T keys.

Cipher 1: key $\mapsto$ many subkeys.

Cipher 2: subkey $\mapsto$ outputs.

A simple tight new proof

Remember the goal: analyze p_T.

There are T keys.

Cipher 1: key $\mapsto$ many subkeys.

Cipher 2: subkey $\mapsto$ outputs.

New proof has just two steps.

A simple tight new proof

Remember the goal: analyze p_T.

There are T keys.
Cipher 1: key $\mapsto$ many subkeys.
Cipher 2: subkey $\mapsto$ outputs.

New proof has just two steps.

Step 1. Replace all subkeys.
Distinguisher $\Rightarrow$ T-target
attack against cipher 1.

A simple tight new proof

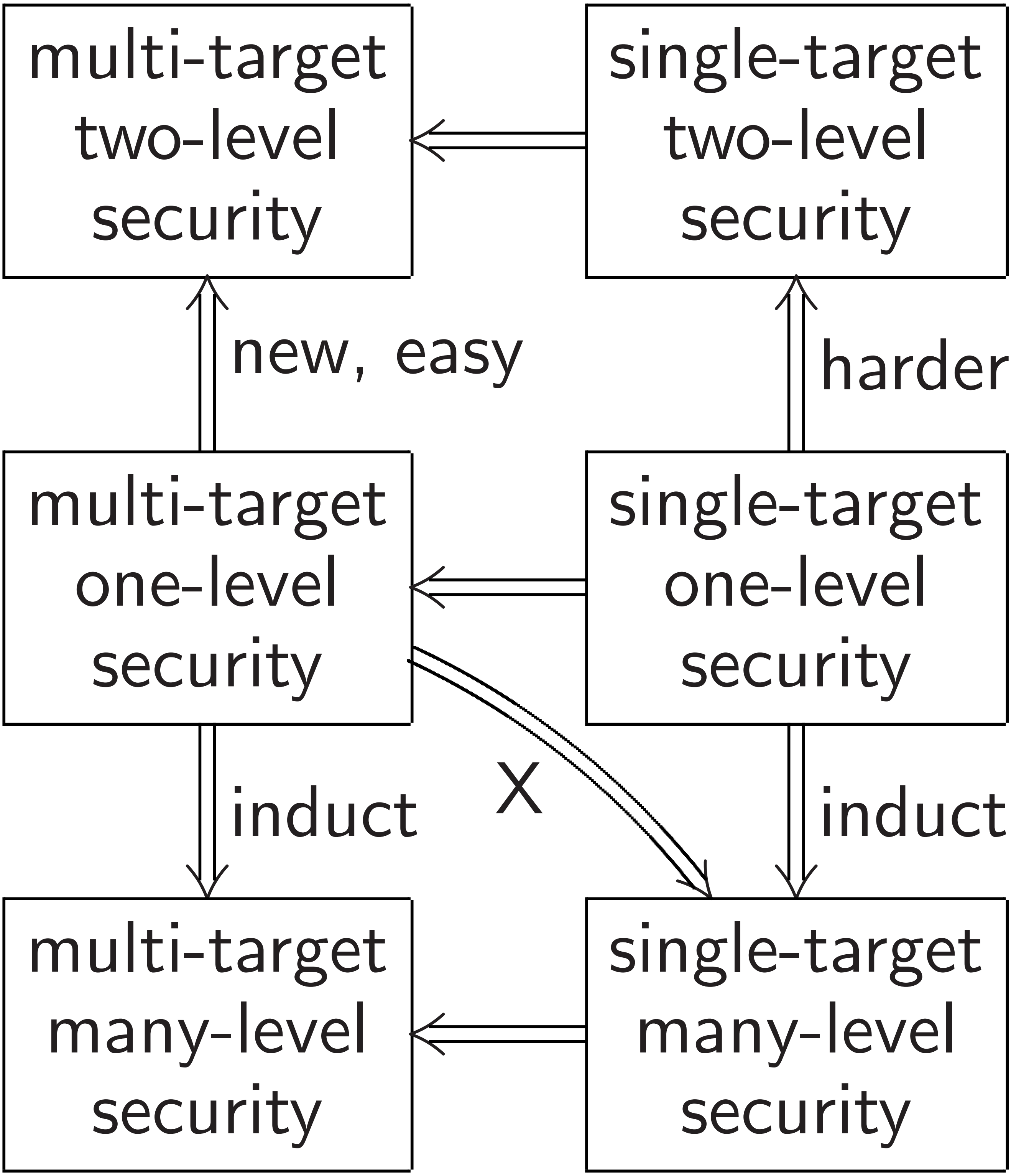
Remember the goal: analyze p_T.

There are T keys.
Cipher 1: key $\mapsto$ many subkeys.
Cipher 2: subkey $\mapsto$ outputs.

New proof has just two steps.

Step 1. Replace all subkeys.
Distinguisher $\Rightarrow$ T-target
attack against cipher 1.

Step 2. Replace all outputs.
Distinguisher $\Rightarrow$ $(T \cdot \text{many})$-target
attack against cipher 2.



X: FOCS 1996 Bellare–Canetti–Krawczyk Lemma 3.2. Harder; not suitable for induction.