Symmetric-key cryptography V

Example of block cipher

Tanja Lange
(with lots of slides by Daniel J. Bernstein)

Eindhoven University of Technology

2MMC10 – Cryptology

How do we know that AES is a PRP?

How do we know that AES is a PRP? Answer: we don't!

How do we know that AES is a PRP? Answer: we don't!

Can use proof by reduction to show that AES used in a mode is secure if AES is a PRP but the latter requires cryptanalysis.

This is no different from public-key crypto.

Also we again need a family of block ciphers to make definitions work.

How do we know that AES is a PRP? Answer: we don't!

Can use proof by reduction to show that AES used in a mode is secure if AES is a PRP but the latter requires cryptanalysis.

This is no different from public-key crypto.

Also we again need a family of block ciphers to make definitions work.

Have to check proofs carefully! OCB2 broken in 2018/19 after "proof" and standardization in 2003.

Security depends on block size n; so small PRP is no good.
E.g. counter mode on 4-bit PRP repeats after 16 steps.

How do we know that AES is a PRP? Answer: we don't!

Can use proof by reduction to show that AES used in a mode is secure if AES is a PRP but the latter requires cryptanalysis.

This is no different from public-key crypto.

Also we again need a family of block ciphers to make definitions work.

Have to check proofs carefully! OCB2 broken in 2018/19 after "proof" and standardization in 2003.

Security depends on block size n; so small PRP is no good.
E.g. counter mode on 4-bit PRP repeats after 16 steps.

Even somewhat larger n is dangerous.
DES and Triple-DES use $n = 64$. 2016 Bhargavan–Leurent `sweet32.info` break use of Triple-DES in TLS.

How do we know that AES is a PRP? Answer: we don't!

Can use proof by reduction to show that AES used in a mode is secure if AES is a PRP but the latter requires cryptanalysis.

This is no different from public-key crypto.

Also we again need a family of block ciphers to make definitions work.

Have to check proofs carefully! OCB2 broken in 2018/19 after "proof" and standardization in 2003.

Security depends on block size n; so small PRP is no good.
E.g. counter mode on 4-bit PRP repeats after 16 steps.

Even somewhat larger n is dangerous.
DES and Triple-DES use $n = 64$. 2016 Bhargavan–Leurent `sweet32.info` break use of Triple-DES in TLS.

AES standardized with $n = 128$. More recent designs (sponges, internal state in stream cipher ChaCha20) use ≥ 512.

How do we know that AES is a PRP? Answer: we don't!

Can use proof by reduction to show that AES used in a mode is secure if AES is a PRP but the latter requires cryptanalysis.

This is no different from public-key crypto.

Also we again need a family of block ciphers to make definitions work.

Have to check proofs carefully! OCB2 broken in 2018/19 after "proof" and standardization in 2003.

Security depends on block size n; so small PRP is no good.
E.g. counter mode on 4-bit PRP repeats after 16 steps.

Even somewhat larger n is dangerous.
DES and Triple-DES use $n = 64$. 2016 Bhargavan–Leurent `sweet32.info` break use of Triple-DES in TLS.

AES standardized with $n = 128$. More recent designs (sponges, internal state in stream cipher ChaCha20) use ≥ 512.

Upcoming slides: mostly good cipher apart from too small $n = 64$.
But: small variations can be quickly broken.

TEA, a tiny encryption algorithm, $n = 64$, $|k| = 128$

1994 Wheeler and Needham

```
void encrypt(uint32 *b,uint32 *k)
{
  uint32 x = b[0], y = b[1];
  uint32 r, c = 0;
  for (r = 0;r < 32;r += 1) {
    c += 0x9e3779b9;
    x += y+c ^ (y<<4)+k[0]
             ^ (y>>5)+k[1];
    y += x+c ^ (x<<4)+k[2]
             ^ (x>>5)+k[3];
  }
  b[0] = x; b[1] = y;
}
```

`uint32`: 32 bits $(b_{31}, b_{30}, \ldots, b_1, b_0)$ representing integer $2^{31}b_{31} + 2^{30}b_{30} + \cdots + 2b_1 + b_0$.

+: addition mod 2^{32}.

`c += d`: same as `c = c + d`.

`0x9e3779b9`: hexadecimal for 2654435769.

^: xor; $\oplus$; addition of each bit separately mod 2. Lower precedence than + in C (= first do + operation), so spacing is not misleading.

<<4: multiplication by 16, i.e., $(b_{27}, \ldots, b_1, b_0, b_0, 0, 0, 0, 0)$.

>>5: division by 32, i.e., $(0, 0, 0, 0, 0, b_{31}, \ldots, b_6, b_5)$.

TEA, a tiny encryption algorithm, $n = 64, |k| = 128$

1994 Wheeler and Needham

```
void encrypt(uint32 *b,uint32 *k)
{
  uint32 x = b[0], y = b[1];
  uint32 r, c = 0;
  for (r = 0;r < 32;r += 1) {
    c += 0x9e3779b9;
    x += y+c ^ (y<<4)+k[0]
             ^ (y>>5)+k[1];
    y += x+c ^ (x<<4)+k[2]
             ^ (x>>5)+k[3];
  }
  b[0] = x; b[1] = y;
}
```

Right column: Diagram for one of the 32 rounds, without update of c.

Image credit: adapted from Roberto Avanzi