Cryptography Basics

Symmetric Key Cryptography

Prof.dr. Ferucio Laurenţiu Ţiplea

Fall 2023

Department of Computer Science
"Alexandru Ioan Cuza" University of Iaşi
Iaşi 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Outline

Symmetric cryptography

Symmetric cryptography includes:

- Symmetric encryption;
- Hash functions;
- Message authentication codes;
- Authenticated encryption.

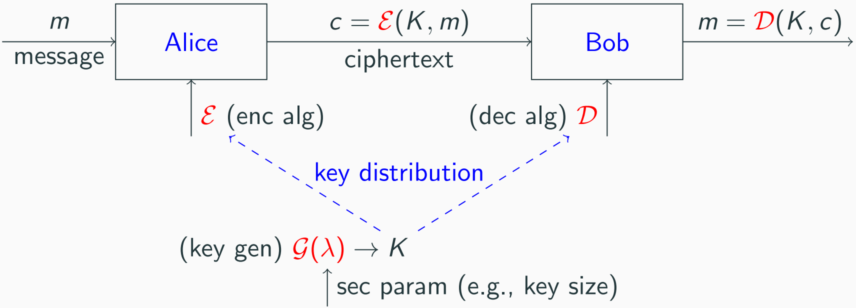
Symmetric encryption

Symmetric encryption

Two main goals:

1. Secure communication
   - protects data in motion;
   - IPsec and SSL&TLS use it;
2. File protection
   - protects data at rest;
   - cloud storage tools use it.

Symmetric cipher: $\mathcal{S} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$



Key distribution:

- Alice and Bob meet and get K, or
- Alice and Bob use a dedicated mechanism/protocol.

Security models for symmetric encryption

Recall that a security model is a pair consisting of a security goal X and an attack model Y, usually written as X-Y.

Standard security goals for encryption:

1. Semantic security (SS);
2. Indistinguishability (IND);
3. Non-malleability (NM).

Standard attack models for encryption:

1. Chosen plaintext attack (CPA);
2. Non-adaptive chosen ciphertext attack (CCA1);
3. Adaptive chosen ciphertext attack (CCA2).

Security goals

1. Semantic security
   1.1 Proposed by Goldwasser and Micali (1984), it was the first definition of security for encryption;
   1.2 It formalizes the fact that no adversary can obtain any partial information about the message of a given ciphertext (whatever can efficiently be computed about a message from its ciphertext can also be computed without the ciphertext);
   1.3 It is a "polynomially bounded" version of the concept of perfect secrecy introduced by Shannon (1949);
   1.4 It is complex and difficult to work with;
2. Indistinquishability is an equivalent definition to semantic security which is somewhat simpler;
3. Non-malleability means that, given a ciphertext c of some message m, no efficient adversary can construct another ciphertext c' of some message m' meaningfully related to m.

Attack models

1. Passive attacks:
   1.1 Cipher-only attack (COA): $\mathcal{A}$ has access to the ciphertext;
   1.2 Known plaintext attack (KPA): $\mathcal{A}$ knows pairs (plaintext,ciphertext);
2. Active attacks:
   2.1 Chosen plaintext attack (CPA): $\mathcal{A}$ has access to the encryption oracle (this is for free for PKE);
   2.2 Non-adaptive chosen ciphertext attack (CCA1): $\mathcal{A}$ has, in addition to the ability of a CPA adversary, access to a decryption oracle before the challenge phase;
   2.3 Adaptive chosen ciphertext attack (CCA2): $\mathcal{A}$ has, in addition to the ability of a CCA1 adversary, access to a decryption oracle after the challenge phase. However, no decryption query is allowed involving the challenge ciphertext.

Proving security by indistinguishability

Indistinguishability, as a security model, requires that an adversary from a specific class of adversaries interact with the cryptographic scheme under the security study, as follows:

Training The adversary trains with the scheme according to his type;

Challenge At a given moment, the adversary will choose 2 (different) messages of equal length and will receive the ciphertext of one of them (chosen randomly uniformly);

Training Depending on the class to which the adversary belongs, he can still train with the scheme;

Guess The adversary will have to decide from which of the two messages the ciphertext comes.

If the guessing probability is non-negligible greater than $1/2$, then the adversary wins the game, which means that the scheme is not secure for adversaries in this class.

Relationships among security models

1. IND-COA (also called indistinguishability in the presence of an eavesdropper) is the weakest form of security where the adversary can only eavesdrop on ciphertexts;
2. IND-KPA (also called indistinguishability under multiple encryption attack) is stronger than IND-COA;
3. The diagram below only aims to create an image on the relationships between the other security models (some of these relationships are far from trivial).

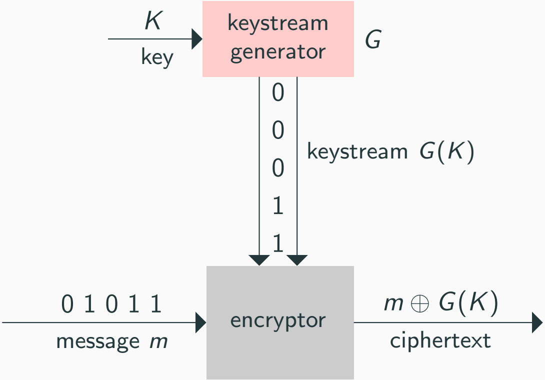
$$
\begin{tikzcd}
\text{NM-CPA} \arrow[d] \arrow[dr, "/" marking, shift left] & \text{NM-CCA1} \arrow[l] \arrow[d] \arrow[r, "/" marking, shift right] & \text{NM-CCA2} \arrow[l, shift right] \arrow[d, shift right] \\
\text{IND-CPA} \arrow[d, shift right] & \text{IND-CCA1} \arrow[l] \arrow[ul, "/" marking, shift left] \arrow[d, shift right] & \text{IND-CCA2} \arrow[l] \arrow[u, shift right] \arrow[d, shift right] \\
\text{SS-CPA} \arrow[u, shift right] & \text{SS-CCA1} \arrow[l] \arrow[u, shift right] & \text{SS-CCA2} \arrow[l] \arrow[u, shift right]
\end{tikzcd}
$$

Stream ciphers

Main characteristics of a stream cipher:

- The message is viewed as a sequence of blocks (also called characters) of a very limited size, that can efficiently be enumerated in practice (e.g., bits or bytes);
- The secret key is expanded to a keystream of the same size as the message block size by a keystream generator initially seeded with the secret key;
- The encryption is block-driven ;
- One-time pad (OTP) may be regarded as a stream cipher, but a quite impractical one.

Stream ciphers



Theorem 1

The SKE scheme above is IND-COA, provided that G is a PRG.

Stream ciphers: using the same key twice

Using the same key twice:

- If $c_1 = m_1 \oplus G(K)$ and $c_2 = m_2 \oplus G(K)$, then $c_1 \oplus c_2 = m_1 \oplus m_2$;
- Natural language text contains enough redundancy to allow the adversary to recover m_1 and m_2 from $c_1 \oplus c_2$.

Real scenarios:

- Microsoft implementation of PPTP in Windows NT uses RC4. Its original implementation uses the same key to encrypt messages from A to B and from B to A (see ScMu1998.pdf on the course site);
- Microsoft have used RC4 to protect Word and Excel document. When encrypted documents were modified and saved, the same key was used (see Wu2005.pdf on the course site).

Never use the same key to encrypt more than one message with stream ciphers !

Stream ciphers: malleability

Malleability:

- From an encryption $c = m \oplus G(K)$ of m one can simply obtain an encryption of $m \oplus m'$ by $c' = c \oplus m'$.

Real scenarios:

- Assume that the adversary knows a prefix m_1 of m (m_1 might be a standard header filled with someone's address, name, etc.);
- The adversary wants to replace m_1 by m_2 (m_2 might be a header filled with information up to his desire);
- The adversary may compute $c \oplus (m_1 \oplus m_2)0 \cdots 0$ to obtain what he wants.

Stream ciphers do not guarantee integrity !

The stream cipher RC4

1. RC4 was proposed by Ronald Rivest in 1987 as a trade secret but posted anonymously in September 1994 on a mailing list;
2. RC4 was used in a large variety of applications: SSL/TLS, WEP, WPA, MS-PPTP etc.;
3. Recent results have shown that the *RC4_gen* output is biased (see AlFardan et al. (2013)):
   3.1 (Mantin & Shamir, 2001) $P(Z_2 = 0x00) \approx \frac{1}{128}$;
   3.2 (Gupta et al., 2012) $P(Z_r = 0x00) \approx \frac{1}{256} + \frac{c_r}{256^2}$ for $3 \leq r \leq 255$, where $c_3 = 0.351089$ and $0.242811 \leq c_r \leq 1.337057$ for $r \geq 4$;
4. Several other variants of RC4 have been proposed: *RC4A*, *VMPC*, $RC4^+$, *Spritz*.

Other practical stream ciphers

1. CSS (Content Scrambling System)
   - Designed in 1980's for preventing unauthorized duplication of DVDs;
   - Can be brute-force attacked in time 2^{40} (the seed space size). A faster attack to recover the seed (time 2^{16}) was proposed by Frank Stevenson in 1999;
2. A5/1, A5/2, A5/3 stream ciphers for GSM encryption
   - All have been cryptanalysed (see Barkan et al. (2003));
3. E0 stream cipher for Bluetooth encryption
   - The most efficient cryptanalysis requires the first 24 bits of $2^{23.8}$ frames (a frame is 2745 bits long) and 2^{38} computations to recover the key (see Lu et al. (2005));
4. Salsa, designed by Bernstein in 2005 (see Bernstein (2008b));
5. ChaCha, designed by Bernstein in 2008 (see Bernstein (2008a)).

Block ciphers

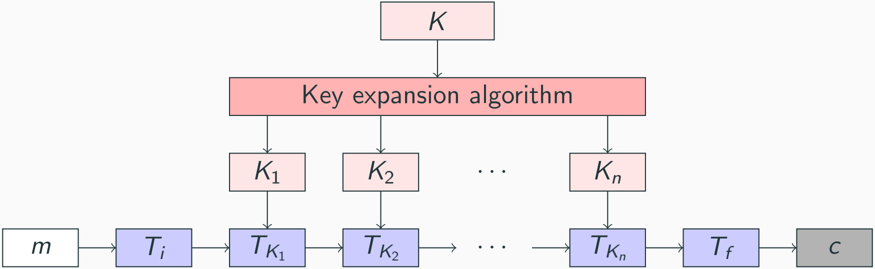
An intensively used method to encrypt a message is the next one:

1. View the message as a sequence of blocks of a larger size so that the enumeration of all blocks is infeasible in practice;
2. Iteratively encrypt each message block by another block.

Remark 2

1. *The encryption of a message block by another block is done by families of permutations (i.e., block ciphers) or families of functions;*
2. *The iteration method is crucial and it is called mode of operation;*
3. *In the encryption process of a message block, the encryption key is expanded to a fixed number of round keys.*

Block ciphers



- T_i is an initial transformation, and T_f is a final transformation;
- T_{K_i} is a transformation induced by K_i, $1 \leq i \leq n$

$$c = (T_f \circ T_{K_n} \circ \cdots \circ T_{K_1} \circ T_i)(m)$$

DES and AES block cipher

1. DES:
   - $\mathcal{M} = \mathcal{C} = \{0,1\}^{64}$;
   - $\mathcal{K} = \{0,1\}^{56}$;
   - The number of rounds is 16;
2. AES:
   - $\mathcal{M} = \mathcal{C} = \mathcal{M}_{4\times m}(\mathbb{Z}_2^8)$, where $m \in \{4,6,8\}$;
   - $\mathcal{K} = \mathcal{M}_{4\times k}(\mathbb{Z}_2^8)$, where $k \in \{4,6,8\}$;
   - The number of rounds varies on the key and message block length

	$m = 4$	$m = 6$	$m = 8$
$k = 4$	10	12	14
$k = 6$	12	12	14
$k = 8$	14	14	14

Pseudo-random functions

A pseudo-random function (PRF) is a family $\mathcal{F}$ of functions with the following properties:

1. Efficiently computable: Each function $f \in \mathcal{F}$ can be computed by a deterministic poly-time algorithm;
2. Pseudo-randomness: If we randomly choose a function from this family then its input-output behavior is computationally indistinguishable from that of a random function.

The adversary is allowed to train with $f \in \mathcal{F}$ to establish the pseudo-randomness of f (see "indistinguishability")!

Pseudo-random permutations

Pseudo-random permutations (PRPs) are special cases of PRFs.

$\mathcal{F}$	Content type	Efficient computability	Pseudo-randomness
PRF	functions	each f	$\mathcal{A}$ trains with f
weak PRP	permutations	each f and f^{-1}	$\mathcal{A}$ trains with f
strong PRP	permutations	each f and f^{-1}	$\mathcal{A}$ trains with f and f^{-1}

Remark 3

1. *Strong PRP are simply referred to as PRP;*
2. *PRP are sometimes called block ciphers.*

PRP candidates

1. $DES = (DES_K)_{K\in\{0,1\}^{56}}$, where

$$DES_K : \{0,1\}^{64} \to \{0,1\}^{64}$$

2. $3DES = (3DES_K)_{K\in\{0,1\}^{168}}$, where

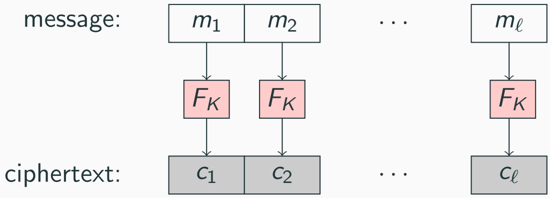
$$3DES_K : \{0,1\}^{64} \to \{0,1\}^{64}$$

3. $AES\text{-}128 = (AES_K)_{K\in\{0,1\}^{128}}$, where

$$AES_K : \{0,1\}^{128} \to \{0,1\}^{128}$$

Electronic Code Block (ECB)

$F = (F_K)_K$ is a PRP

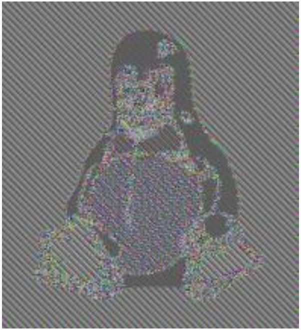


Theorem 4

ECB is not IND-KPA.

ECB illustrated

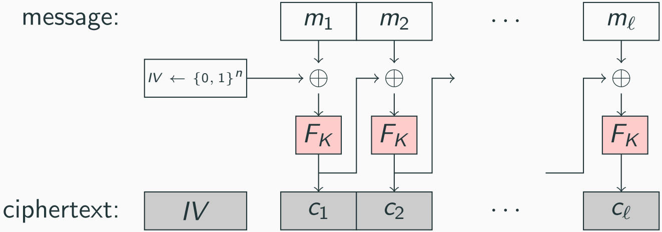
Original image



ECB encryption

Cipher Block Chaining (CBC)

$F = (F_K)_K$ is a PRP

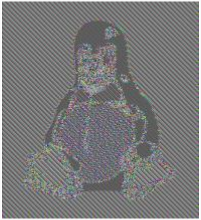


Theorem 5

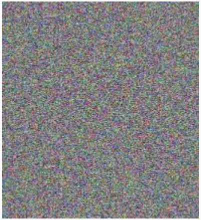
If $F = (F_K)_K$ is a PRP, then CBC with F is IND-CPA.

CBC versus ECB

Original image



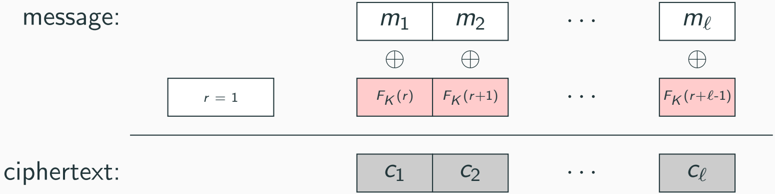
ECB encryption



CBC encryption

Deterministic counter mode (DCTR)

$F = (F_K)_K$ is a PRF



The scheme works like a stream cipher with the PRG G given by

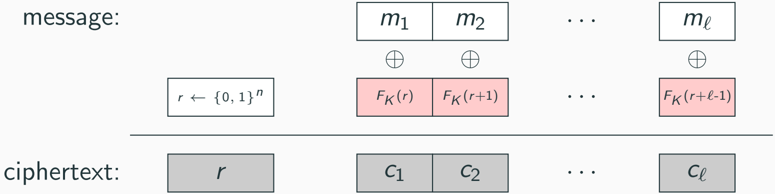
$$G(K) = F_K(1) \parallel F_K(2) \parallel \cdots \parallel F_K(\ell)$$

Theorem 6

If $F = (F_K)_K$ is a PRF, then DCTR with F is IND-KPA but not IND-CPA.

Counter mode (CTR)

$F = (F_K)_K$ is a PRF



The scheme works like a stream cipher with the PRG G given by

$$G(K) = F_K(r) \parallel F_K(r+1) \parallel \cdots \parallel F_K(r+\ell-1)$$

Theorem 7

If $F = (F_K)_K$ is a PRF, then CTR with F is IND-CPA.

Output feedback (OFB) and cipher feedback (CFB)

1. The key stream in CTR mode is

$$F_K(r) \parallel F_K(r+1) \parallel F_K(r+2) \parallel \cdots$$

   where $r \leftarrow \{0,1\}^n$

2. The OFB and CFB modes are defined as the CTR mode but with a different key stream generation :

   2.1 The key stream in OFB mode is

   $$F_K(r) \parallel F_K(F_K(r)) \parallel F_K(F_K(F_K(r))) \parallel \cdots$$

   where $r \leftarrow \{0,1\}^n$

   2.2 The key stream in CFB mode is

   $$F_K(r) \parallel F_K(c_1) \parallel F_K(c_2) \parallel \cdots$$

   where $r \leftarrow \{0,1\}^n$

Hash functions

Hash functions

A hash function outputs a fixed-length bitstring (e.g., 128 or 160) when applied to an arbitrary-length bitstring.

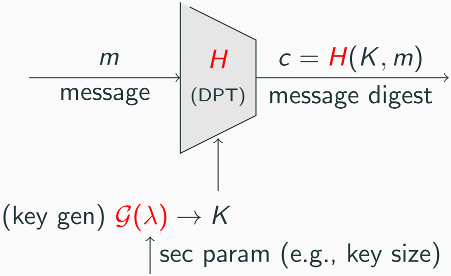
Hash functions are used in many cryptographic applications such as:

- signing messages, in connection with digital signatures (signing a document should be a fast operation and the signature should be small so that it can be put on a smart card);
- identifying files on peer-to-peer file sharing networks;
- ensuring security of micro-payment schemes (e.g., PayWord);
- etc.

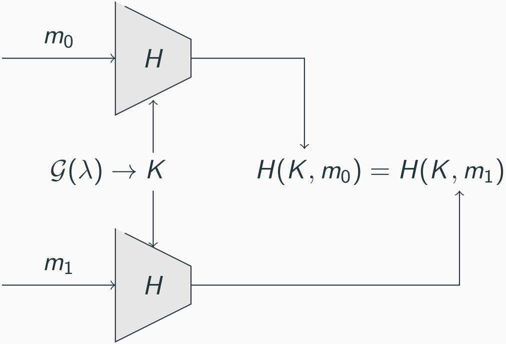
Hash functions

(Keyed) hash function: $\mathcal{H} = (\mathcal{G}, H)$

When no key is used, H is called a hash function.



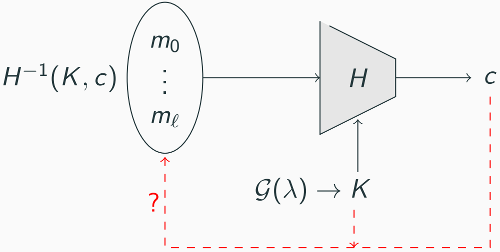
Collision-resistant hash functions



If $m_0 \neq m_1$ then (m_0, m_1) is a collision

A keyed hash function $\mathcal{H}$ is collision-resistant (CRHF) if no adversary, given a randomly generated key K, can compute a collision (m_0, m_1) for H under K with a higher than negligible probability.

One-way hash functions



A keyed hash function $\mathcal{H}$ is one-way (OWHF) if no adversary, given a randomly generated key K and a message digest c obtained with K, can compute $m \in H^{-1}(K, c)$ with a higher than negligible probability.

Theorem 8

Any CRHF is also a OWHF, as long as the domain of the hash function is significantly larger than its range.

Looking for collisions

Theorem 9

Let m be the number of possible message digests of a hash function H under some key K. If we compute message digests for r messages chosen uniformly at random and

$$\lfloor \sqrt{2cm} \rfloor < r < m$$

for some real constant $c > 0$, then the probability to get a collision is higher than $1 - e^{-c}$ (e is Euler's number, $e = 2.71828 \cdots$).

If $c \geq \ln 2 \sim 0.693$, then $1 - e^{-c} > \frac{1}{2}$

Example 10

Let $m = 2^{40}$ and r such that $1.200.000 \approx \lfloor 2^{20} \sqrt{2 \ln 2} \rfloor < r < 2^{40}$.

The probability of getting a collision is greater than $1/2$. Therefore, 40-bit message digests do not ensure security.

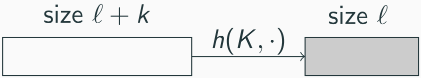
Construction of CRHFs

Two practical techniques to construct CRHFs:

1. The Merkle-Damgard (MD) transform;
2. The sponge construction.

The MD transform

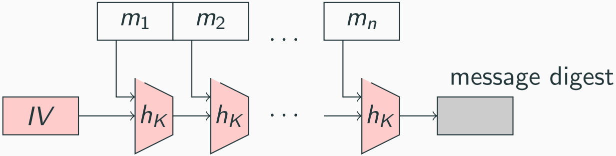
- Use a compression function $h : \mathcal{K} \times \{0,1\}^{\ell+k} \rightarrow \{0,1\}^{\ell}$



- Use an MD-complaint padding $pad : \{0,1\}^{<2^{\ell}} \rightarrow \bigcup_{n\geq 1}\{0,1\}^{n\ell}$ with the following properties:
  1. m is a prefix of $pad(m)$
  2. if $|m_1| = |m_2|$ then $|pad(m_1)| = |pad(m_2)|$;
  3. if $m_1 \neq m_2$, then the last block of $pad(m_1)$; is different than the last block of $pad(m_2)$;

The MD transform

- Iterate h on messages m as follows:
  1. $pad(m) = m_1 \parallel \cdots \parallel m_n$ with $|m_i| = k$ for all i;
  2. $V := IV$, where $IV \leftarrow \{0,1\}^{\ell}$;
  3. for $i := 1$ to n do $V := h(K, m_i \parallel V)$;
  4. return V;



Theorem 11

If h is collision-resistant, then the MD-transform based on h is so.

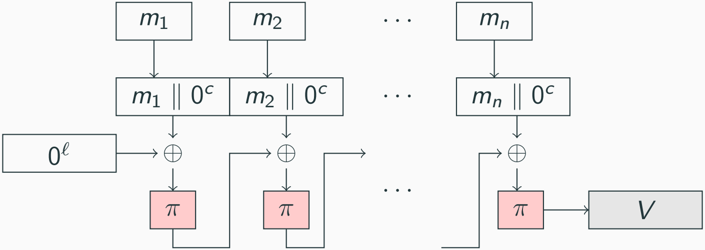
The MD transform in practice

Practical hash functions based on the MD-transform:

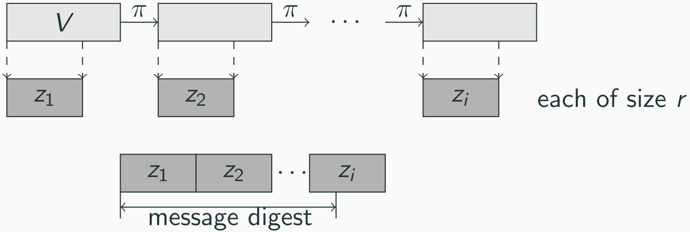
- MD4 – developed by Rivest in 1990. It was the starting point for the development of a series of similar hash functions;
- SHA (Secure Hash Algorithm) or SHA-0 – developed by NSA in 1993 (withdrawn shortly after publication because of some flaw);
- MD5 – the strengthened successor of MD4 (Rivest 1995);
- SHA-1 – developed by NSA in 1995; not longer approved after 2010;
- SHA-2 family includes 6 hash functions, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 (the last two are truncated versions of SHA-512).

The sponge construction

- Choose a permutation $\pi : \{0,1\}^\ell \to \{0,1\}^\ell$ (π has no key!); and write $\ell = r + c$ (r is the rate and c is the capacity);
- Pad m and divide it into r-bit blocks $m_1 \cdots m_n$;
- Absorbing phase.



- Squeezing phase



Theorem 12

If π is a random permutation and 2^ℓ and 2^c are super-poly, then the sponge construction yields a CRHF.

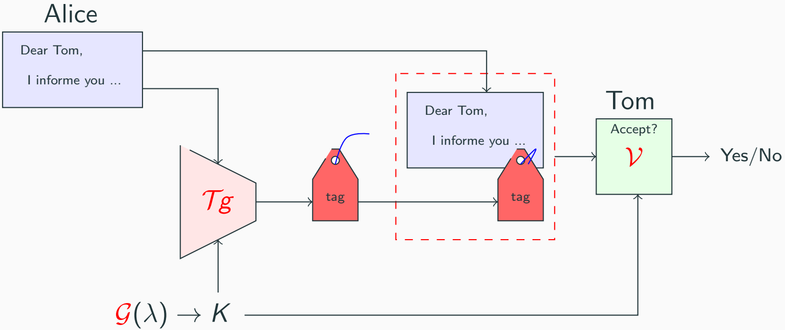
The sponge construction is the basis of SHA-3 standard.

Message integrity

Message authentication codes

Message authentication codes (MACs) = used to prove message integrity based on a shared secret key between parties

MAC system: $\mathcal{S} = (\mathcal{G}, \mathcal{T}g, \mathcal{V})$



MACs

A MAC system $\mathcal{S}$ is secure if no adversary, who has been allowed to train with the MAC system, can generate valid tags for messages of his choice, except with negligible probability.

MAC systems can be obtained from:

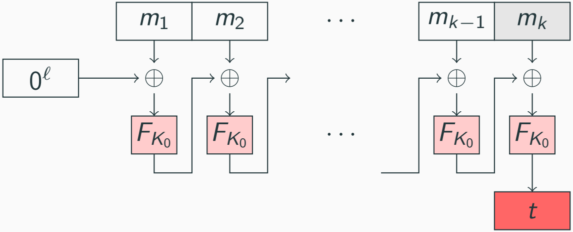
1. PRFs;
2. Hash functions.

MACs from PRFs: CMAC

If F is a PRF on messages of length ℓ, define the following MAC scheme, called Cipher-based MAC (CMAC) (see Dworkin (2005)):

1. Generate three keys K_0, K_1, and K_2 of length ℓ from K;
2. Break the message m into $m = (m_1, \ldots, m_{k-1}, m_k)$;
3. Randomize the last block:
   - If $|m_k| = \ell$ then replace m_k by $m_k \oplus K_1$;
   - If $|m_k| < \ell$ then replace m_k by $(m_k \parallel 1 \parallel 0^j) \oplus K_2$;
4. Apply F with K_0 in the CBC mode and output only the last block as the message tag.

MACs from PRFs: CMAC



Theorem 13

CMAC is secure, provided that F is a PRF.

MACs from CRHFs: HMAC

Let H be a hash function defined by the MD transform from a compression function $h(K, m)$. Define F_H by

$$F_H((K_1, K_2), m) = H(K_2 \parallel H(K_1 \parallel m))$$

Theorem 14

If h and h' given by $h'(K, m) = h(m, K)$ are PRFs, then F_H is a PRF.

For a proof of this theorem please see Boneh and Shoup (2020).

The HMAC construction uses one single key K from which two keys are derived: $K_1 = K \oplus ipad$ and $K_2 = K \oplus opad$.

HMAC-SHA1 and HMAC-SHA256 are instances of the above construction, with $H = SHA1$ and $H = SHA256$, respectively.

Authenticated encryption

The need for authenticated encryption

Combining secure encryption schemes with secure MACs may lead to error-prone systems (see Krawczyk (2001), Bernstein (2013))

Definition 15

Let $\mathcal{S}$ be a cipher.

1. $\mathcal{S}$ provides ciphertext integrity (CI) if no adversary can output valid ciphertexts, except with negligible probability.
2. $\mathcal{S}$ provides authenticated encryption (AE) if:
   2.1 $\mathcal{S}$ is IND-CPA secure
   2.2 $\mathcal{S}$ provides CI.

Theorem 16

If $\mathcal{S}$ is AE secure then it is IND-CCA secure.

Constructing AE secure ciphers

One popular way to construct AE secure ciphers is to combine an IND-CPA secure cipher with a secure MAC. There are two main variants:

1. Encrypt-then-MAC (EtM)
   1.1 $c \leftarrow \mathcal{E}(K, m)$, $t \leftarrow \mathcal{T}g(K', c)$, output (c, t);
   1.2 Used in IPsec, TLS 1.2 and later versions, and in the NIST standard GCM;
2. MAC-then-Encrypt (MtE)
   2.1 $t \leftarrow \mathcal{T}g(K', m)$, $c \leftarrow \mathcal{E}(K, (m, t))$, output c;
   2.2 Used in SSL 3.0, TLS 1.0, and in 802.11i WiFi encryption protocol.

The keys K and K' are chosen independently!

Encrypt-then-MAC

Theorem 17

If $\mathcal{S}$ is an IND-CPA secure cipher and $\mathcal{S}'$ is a secure MAC, then the EtM construction is a secure AE.

Common mistakes in implementing the EtM construction:

1. Use the same key for the cipher and the MAC;
2. Apply the MAC only to part of the ciphertext (we may loose ciphertext integrity) – discovered in 2013 at `RNCryptor` facility in Apple's iOS.

MAC-then-Encrypt

MtE is not generally secure:

1. The attack POODLE on SSL 3.0;
2. Padding oracle timing attack in TLS 1.0;
3. Informative error messages in TLS 1.0.

There are secure instances of MtE:

1. The randomized counter mode of the cipher assures AE security.

References

AlFardan, N., Bernstein, D. J., Paterson, K. G., Poettering, B., and Schuldt, J. C. N. (2013). On the security of rc4 in TLS. In *22nd USENIX Security Symposium (USENIX Security 13)*, pages 305–320, Washington, D.C. USENIX Association.

Barkan, E., Biham, E., and Keller, N. (2003). Instant ciphertext-only cryptanalysis of gsm encrypted communication. In Boneh, D., editor, *Advances in Cryptology - CRYPTO 2003*, pages 600–616, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bernstein, D. J. (2008a). ChaCha, a variant of Salsa20. Technical report, The University of Illinois at Chicago.

Bernstein, D. J. (2008b). The salsa20 family of stream ciphers. In Robshaw, M. and Billet, O., editors, *New Stream Cipher Designs: The eSTREAM Finalists*, pages 84–97, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bernstein, D. J. (2013). Failures of secret-key cryptography. Invited talk to the 20th International Workshop on Fast Software Encryption 2013.

Boneh, D. and Shoup, V. (2020). *A Graduate Course in Applied Cryptography*. Authors' website, Version 0.5.

References (cont.)

Dworkin, M. (2005). Recommendation for block cipher modes of operation: The CMAC mode for authentication. NIST Pubs 800-30B, NIST. Updated, 2016.

Goldwasser, S. and Micali, S. (1984). Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299.

Krawczyk, H. (2001). The order of encryption and authentication for protecting communications (or: How secure is SSL?). In Kilian, J., editor, *Advances in Cryptology — CRYPTO 2001*, pages 310–331, Berlin, Heidelberg. Springer Berlin Heidelberg.

Lu, Y., Meier, W., and Vaudenay, S. (2005). The conditional correlation attack: A practical attack on bluetooth encryption. In Shoup, V., editor, *Advances in Cryptology – CRYPTO 2005*, pages 97–117, Berlin, Heidelberg. Springer Berlin Heidelberg.

Shannon, C. E. (1949). Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4):656–715.