Security Review

Niklas Bühler

Summerterm 2020

Contents

Preface

This is a summary of the most important definitions, theorems and some proofs for the Security lecture at KIT. It is based on the lectures by Prof. Müller-Quade in summer term 2020.

1 General

- Concept of *CIA*: Confidentiality, Integrity, Availability

2 Symmetric Encryption

2.1 One-Time-Pad (OTP)

- Length of key is equal to length of message; $M, K \in \{0,1\}^n$
- Encoding: $E(K, M) = C = M \oplus K \in \{0,1\}^n$
- Decoding: $D(K, C) = C \oplus K = M$
- Important: K has to be chosen at random, uniformly distributed
- $\oplus$ Given C, every possible M is equiprobable
- $\ominus$ The key is bulky, may not be reused
- $\ominus$ Ciphertext is malleable: $C \oplus K = (M \oplus X) \oplus K$

2.2 Stream ciphers

- Idea: Simulate OTP with short $K \in \{0,1\}^k, (k < n)$
- Expand K to $K' := G(K) \in \{0,1\}^n$, then perform OTP using K'
- Goal: pseudorandom number $G(K)$ should "look" truly random
- $\oplus$ Fast, especially in hardware
- $\oplus$ Established construction using multiple linear-feedback shift registers (LFSRs)
- $\ominus$ Oftentimes algebraic attacks possible
- $\ominus$ Requires synchronization for updating key
- $\ominus$ Ciphertext is malleable, like in OTP

2.3 Block ciphers

- $E : \{0,1\}^k \times \{0,1\}^l \to \{0,1\}^l, (K, M) \mapsto C$
- $D : \{0,1\}^k \times \{0,1\}^l \to \{0,1\}^l, (K, C) \mapsto M$
- Correctness: $\forall K, M : D(K, E(K, M)) = M$

2.3.1 Operating modes

2.3.1.1 Electronic Codebook Mode (ECB)

- Idea: Split M into l-bit blocks $M_1, \dots \in \{0,1\}^l$ and let $C := (C_1, \dots)$ with $C_i := E(K, M_i) \in \{0,1\}^l$, decryption works analog
- $\oplus$ Easy to implement, no synchronization required
- $\ominus$ Same M, same C; Insertions or different order possible
- $\ominus$ Bit error in C_i destroys block M_i

2.3.1.2 Cipher Block Chaining Mode (CBC)

- Problem with ECB: cipher blocks are independent $\Rightarrow$ chain them

- Split M into l-bit blocks $M_1, \ldots$
- Let $C_0 := IV$ (initialization vector)
- Let $C_i := E(K, M_i \oplus C_{i-1})$
- Decoding: $M_i := D(K, C_i) \oplus C_{i-1}$
- IV has to be transmitted as well, or be a constant
- $\oplus$ Solves some disadvantages of the ECB: Same message blocks don't result in the same cipher blocks anymore, arranging the cipher blocks in a different order is also not possible anymore
- $\ominus$ Not parallelizable
- $\ominus$ Cipher text is malleable
- $\ominus$ Bit error in C_i at position j destroys block M_i and flips bit j in M_{i+1}

2.3.1.3 Counter Mode (CTR)

- Similar to stream ciphers
- $C_0 := IV, C_i := E(K, IV + i) \oplus M_i$
- Similar properties to CBC (but can be parallelized better)
- Also allows homomorph malleability
- $\Rightarrow$ Use Galois Counter Mode (GCM), which is authenticated

2.3.1.4 Roundup

- Block ciphers use encription E in blocks
- ECB: "raw" E-function $\Rightarrow$ don't use
- CBC, CTR: better, but only secure against eavesdropping
- GCM: best choice

2.4 Data Encryption Standard (DES)

- Uses Feistel cipher
- Round function F is non-invertable, but E is
- Structurally unbroken (but key is too short)
- Input- and output-permutation are inverse, so $IP = FP^{-1}$
- Decryption uses same Feistel cipher, but F-keys are used in reverse

2.5 2DES

- $K := (K_1, K_2) \in (\{0,1\}^{56})^2$
- $E_{2DES}(K, M) := E_{DES}(K_2, E_{DES}(K_1, M)$
- Not really more secure than DES
- Meet-in-the-middle attack
  - Given: $M, C = E_{2DES}(K, M)$
  - Goal: $K = (K_1, K_2)$
  1. Calculate list of all $C_{K_1'} := E_{DES}(K_1', M)$
  2. Sort list lexicographically (for binary search)
  3. Calculate $C_{K_2} := D_{DES}(K_2', C)$ successively
  4. If $C_{K_2'} = C_{K_1'}$, output (K_1', K_2')

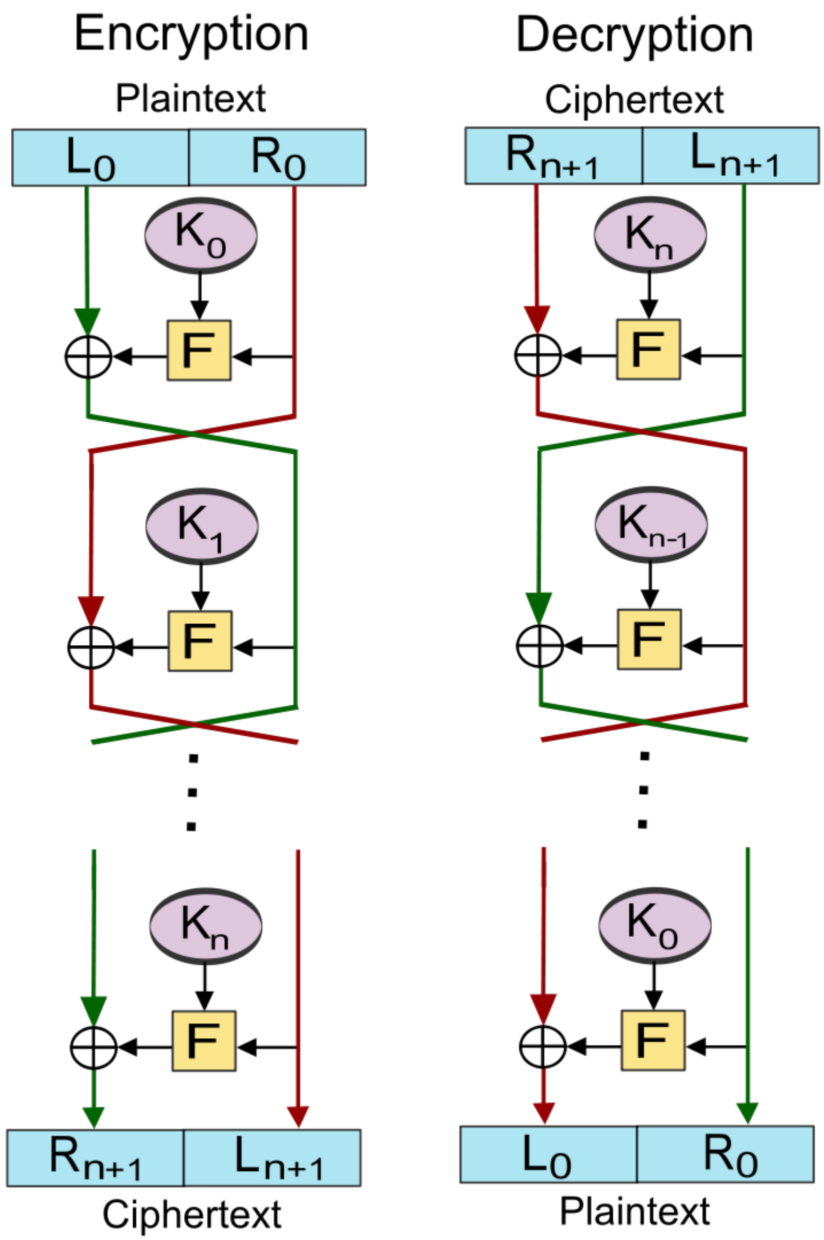

Figure 1: Feistel cipher

2.6 3DES

- Because DES and 2DES are not secure
- $K := (K_1, K_2, K_3) \in (\{0,1\}^{56})^3$
- $E_{3DES}(K, M) := E_{DES}(K_3, D_{DES}(K_2, E_{DES}(K_1, M)))$
- Meet-in-the-middle attack has complexity $\sim 2^{112}$

2.7 Advanced Encryption Standard (AES)

- No Feistel cypher
- According to present knowledge secure

2.8 Linear Cryptanalysis

- Find $\mathbb{F}_2$-linear dependencies in bits of X and $Y := E(K, X)$
  - Enables indirect attack on Feistel cypher (n rounds):
    1. Find linear dependency in F-input and -output
    2. Expand dependency on first $n-1$ rounds
    3. Complete search for last round key $K^{(n)}$
    4. Check $K^{(n)}$ candidate using linear dependency
    5. If $K^{(n)}$ is found, search for $K^{(n-1)}$, $K^{(n-2)}$, ...

2.9 Differential Cryptanalysis

- Consider differences in output $\Delta_{out} := Y \oplus Y'$ in dependence to differences in input $\Delta_{in} := X \oplus X'$
- Attack on Feistel cypher similar to linear cryptanalysis:
  1. Find most probable pairs $\Delta_{in} \Rightarrow \Delta_{out}$ from input and output of second last round
  2. Complete search for last round key $K^{(n)}$, ...
  3. ... check $K^{(n)}$ candidates for consistency of $\Delta_{in} \Rightarrow \Delta_{out}$

2.10 Semantic Security

- Ciphertext does not help with calculations regarding plaintext
- Every information about M that can be calculated (efficiently) with knowledge of C, can also be calculated (efficiently) without knowing the ciphertext
- $\Rightarrow$ only covers passive attacks
- Informal definition: A method of symmetric encryption is semantically secure if for every M-distribution of messages of equal length, every function f and every efficient algorithm A, there exists an efficient algorithm B such that

  $$Pr[A^{Enc(K,\cdot)}(Enc(K, M)) = f(M)] - Pr[B(\epsilon) = f(M)]$$

  is small.
- The existence of (reusable) semantically secure methods implies $P \neq NP$

2.11 Passive Security: IND-CPA

- IND-CPA: Indistinguishability under chosen-plaintext attacks
- Method is IND-CPA-secure $\iff$ there's no efficient attacker A that can distinguish ciphertexts of two chosen plaintexts
  1. A is given access to $Enc(K, \cdot)$ oracle
  2. A chooses two messages $M^{(1)}, M^{(2)}$ of equal length
  3. A receives $C^* := Enc(K, M^{(b)})$ for uniformly distributed $b \in \{1, 2\}$
  4. A wins if it guesses b correctly
- Method is IND-CPA-secure $\iff \forall A : (Pr[A \text{ wins}] - \frac{1}{2})$ is small
- IND-CPA $\iff$ semantic security
- Proofs:
  - Not semantically secure: Build winning A
  - Semantically secure: Use winning A to build something that contradicts the assumptions, e.g. Enc and random discriminator)

3 Hash Functions

3.1 Goals

- Short *fingerprint*: $H : \{0,1\}^* \to \{0,1\}^k$
- Efficient algorithm $H(X)$
- Surjective: $H(\{0,1\}^*) = \{0,1\}^k$
- Avoid collisions, mapping on $\{0,1\}^k$ is uniformly distributed
- Creates *chaos*

3.2 Requirements for a hash function

- *Collision resistance*: hard to find $X \neq X'$ with $H(X) = H(X')$
- *One-way property*: given $Y = H(X)$, X' with $H(X') = Y$ is hard to find
- *Target collision resistance*: given X, X' with $X \neq X'$ and $H(X) = H(X')$ is hard to find

3.3 Collision Resistance (informal)

- Collision: $X_0, X_1 \in \{0,1\}^*$ with $X_0 \neq X_1 \land H(X_0) = H(X_1)$
- Collision resistant $\iff$ every efficient algorithm finds a collision only with small probability

3.4 Trivial Collisionfinder (Brute Force)

- Calculate $H := \{H(X) | X \in \{0,1\}^k\}$ in $O(2^k)$ time
- If no collision is found, then $H(X^*)$ is collision with an $X \in \{0,1\}^k$ for all $X^* \notin \{0,1\}^k$
- Better (in $O(2^{k/2})$ time):
  1. Randomly choose $2^{k/2}$ messages $X_1, \ldots, X_{2^{k/2}}$

2. For $i = 1, \ldots, 2^{k/2}$, calculate $Y_i := H(K_i)$
3. Look for collision $Y_i = Y_j$, if there's none go to 1.
– Approximately 2 iterations needed

3.5 Security Parameter: Asymptotic Definition

- $k \in \mathbb{N}$ parameterizes the system
- *Efficient*: Polynomial time (in k): PPT
- *Small probability*: negligible (in k)
  - $f : \mathbb{N} \to \mathbb{R}$ negligible $\iff$ $|f|$ vanishes asymptotically faster than the reciprocal of every given polynomial
  - $\forall c \exists k_0 \forall k \geq k_0 : |f(k)| \leq k^{-c}$

3.6 Collision Resistance (formal)

A function H that is parameterized by k is *collision resistant* if for every PPT algorithm A

$$Adv^{cr}_{H,A}(k) := Pr[(X, X') \leftarrow A(1^k) : X \neq X' \wedge H_k(X) = H_k(X')]$$

is negligible.

3.7 One-way function

A function H that is parameterized by k is a *one-way function* regarding the distribution $\{\chi_k\}_k$ of the inverse image if for every PPT algorithm A

$$Adv^{ow}_{H,A}(k) := Pr[X' \leftarrow A(1^k, H(X)) : H_k(X) = H_k(X')]$$

is negligible, where $X \leftarrow \chi_k$.

3.8 Theorem: Collision Resistance ⇒ One-way property

Every collision resistant hashfunction $H : \{0,1\}^* \to \{0,1\}^k$ is a one-way function regarding the uniform distribution on $\{0,1\}^{2k}$.

Proof:
For every H-inverter A, there's a H-collision-finder B with

$$Adv^{cr}_{H,B}(k) \geq \frac{1}{2} Adv^{ow}_{H,A}(k) - \frac{1}{2}^{k+1}$$

3.9 Merkle-Damgård Construction

- Build hashfunction H_{MD} out of simpler compression function $F : \{0,1\}^{2k} \to \{0,1\}^k$

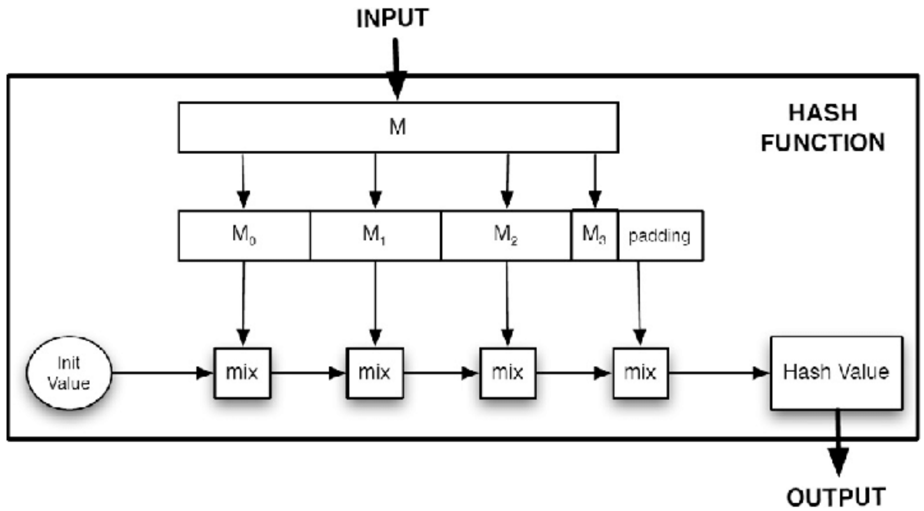

Figure 2: Merkle-Dåmgard construction

3.10 Theorem: F collision resistant $\Rightarrow$ H_{MD} collision resistant

Proof: Given $X \neq X', H_{MD}(X) = H_{MD}(X')$, find F collision

1. Let $X = X_1 \dots X_n, X' = X'_1 \dots X'_n$ with $X_i, X'_i \in \{0,1\}^k$,
   MD intermediate values $Z_0 := IV, Z_i := F(Z_{i-1}, X_i)$
2. $Z_n = F(Z_{n-1}, X_n) = F(Z'_{n'-1}, X'_{n'}) = Z'_{n'}$
3. $Z_{n-1} \neq Z'_{n'-1}$ or $X_n \neq X'_{n'} \Rightarrow F$ collision

Thus, $X_n = X'_{n'}$ and $Z_{n-1} = F(Z_{n-2}, X_{n-1}) = F(Z'_{n'-2}, X'_{n'-1}) = Z'_{n'-1}$, but because of $X \neq X'$, we can't have $Z_i = Z'_i \forall i$. So there'd be an F collision.

4 Symmetric Authentication of Messages

- Goal: authenticated transmission over unauthenticated channel $\rightarrow$ send message M with signature σ
- Requirements:
  - σ can be calculated by sender and verified by receiver
  - Length of σ is small
  - Outsider can't create valid σ for new M

4.1 MACs

- A and B share a secret K
- Signing: $\sigma \leftarrow Sig(K, M), M \in \{0,1\}^*, \sigma \in \{0,1\}^k$

- Verifying: $Ver(K, M, \sigma) \in \{0, 1\}$
- Correctness: $Ver(K, M, \sigma) = 1 \forall K, M$ and $\sigma \leftarrow Sig(K, M)$

4.2 EUF-CMA Security

No PPT-attacker A wins the following game non-negligible often:

1. A is granted access to a $Sig(K, \cdot)$-oracle
2. A outputs (M^*, σ^*)
3. A wins, iff. $Ver(K, M^*, \sigma^*) = 1$ and M^* hasn't been passed to the oracle before

4.3 Theorem: Hash-Then-Sign Paradigm

- Given: (Sig, Ver) EUF-CMA secure and H is a collision resistant hash-function
- Then: MAC $Sig'(K, M) = Sig(K, H(M)), Ver'(K, M, \sigma) = Ver(K, H(M), \sigma)$ is also EUF-CMA secure
- Proof: Any EUF-CMA attacker A' on (Sig', Ver') must either find a H collision or a signature σ for a fresh $H(M)$.

4.4 Preudorandom function PRF

- $PRF : \{0, 1\}^k \times \{0, 1\}^k \rightarrow \{0, 1\}^k$ over $k \in \mathbb{N}$ parameters
- PRF is called a pseudorandom function iff. for ever PPT alorithm A

$$Adv^{prf}_{PRF,A}(k) := Pr[A^{PRF(K,\cdot)}(1^k) = 1] - Pr[A^{R(\cdot)}(1^k = 1]$$

  is negligible, where $R : \{0, 1\}^k \rightarrow \{0, 1\}^k$ is a real random function.

4.5 Creating PRF candidates from hashfunctions

- $PRF(K, X) := H(K||X)$
- Sometimes (Merkle-Dåmgard), a hashvalue is extensible: $H(K||X) = H(K||X||X')$ breaks PRF property for inputs of variable length

4.6 Theorem: MACs from PRFs and hashfunctions

- Given: $PRF : \{0, 1\}^k \times \{0, 1\}^k \rightarrow \{0, 1\}^k$ a PRF and $H : \{0, 1\}^* \rightarrow \{0, 1\}^k$ a collision resistant hashfunction
- Then: $Sig(K, M) = PRF(K, H(M))$ is EUF-CMA secure
- Proof: Assume A to be a succesful EUF-CMA attacker
  - Then A produces fake (M^*, σ^*) with *fresh* M^*
  - A thus represents a PRF-distinguisher that predicts $PRF(K, H(M^*))$

4.7 HMAC

- $Sig(K, M) = H((K \oplus opad)||H((K \oplus ipad)||M))$
- Advantages to $Sig(K, M) = H(K||H(M))$:
  - Additional parameterization makes attacks harder
  - H collisions don't necessarily lead to breakage of Sig

4.8 CBC-MAC: MAC from CBC-Mode

- Choose IV and pick last block of ciphertext as MAC
- If message is encrypted by CBC as well, don't choose the same key!

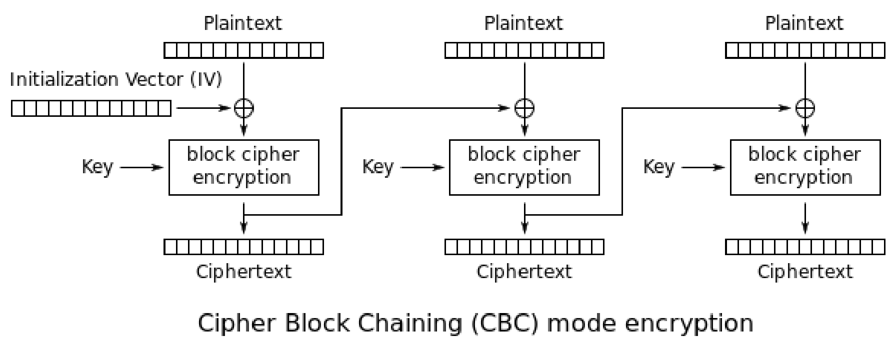

Figure 3: CBC

5 Asymmetric Encryption (Public Key)

- Idea:
  - Encryption using public key: $C \leftarrow Enc(pk, M)$
  - Decryption using secret key: $M \leftarrow Dec(sk, C)$
  - pk and sk are generated together: $(pk, sk) \leftarrow Gen(1^k)$
  - pk is public, sk is secret
  - Thus, there is no (secret) key distribution, for n users there are only n public and n secret keys
- It's often good to use hybrid methods: asymmetric method to transfer key K and afterwards a symmetric method using K

5.1 RSA

- $pk = (N, e), sk = (N, d)$
- $N = PQ$ for (sufficiently large) primes $P \neq Q$
- Calculate in $\mathbb{Z}/N\mathbb{Z}$, where e and d are inverse exponents:
  - $e \cdot d \equiv 1 \mod \varphi(N)$ with $\varphi(N) = (P-1)(Q-1)$
- Message room is $\mathcal{M} := \mathbb{Z}_N$

- $Enc(pk, M) = M^e \mod N$
- $Dec(sk, C) = C^d \mod N$

5.1.1 RSA Key Generation

- Goal: $pk = (N, e), sk = (N, d)$
- Gen chooses P and Q of given bit length k randomly
  - e.g. choose uniformly distributed uneven $P \in \{2^k, \ldots, 2^{k+1}\}$ until P is prime
- To get e and d:
  - Choose uniformly distributed $e \in \{3, \ldots, \varphi(M) - 1\}$ until $gcd(e, \varphi(N)) = 1$
  - Calculate $d = e^{-1} \mod \varphi(N)$ using the extended Euclidean algorithm:
    * $EE(e, \varphi(N)) = (\alpha, \beta)$ with $\alpha e + \beta\varphi(N) = gcd(e, \varphi(N)) = 1$
    * Then $\alpha e = 1 \mod \varphi(N)$, so set $d := \alpha \mod \varphi(N)$

5.1.2 Correctness of RSA

We have to prove $(M^e)^d \equiv M^{ed} \equiv M \mod N$.

5.1.2.1 Theorem: Fermat's little theorem

For prime P and $M \in \{1, \ldots, P-1\}$ we have $M^{P-1} \equiv 1 \mod P$.
Thus, $\forall M \in \mathbb{Z}_P, \alpha \in \mathbb{Z} : (M^{P-1})^\alpha \cdot M \equiv M \mod P$.

5.1.2.2 Theorem: Chinese remainder theorem

Let $N = PQ$, where P and Q are coprime. Then $\mu : \mathbb{Z}_N \to \mathbb{Z}_P \times \mathbb{Z}_Q$ with $\mu(M) = (M \mod P, M \mod Q)$ is bijective.
Thus, $(X \equiv Y \mod P) \wedge (X \equiv Y \mod Q) \Rightarrow X \equiv Y \mod N$.

5.1.2.3 Proof

Show: Let N, e, d be defined as above, then $M^{ed} \equiv M \mod N \; \forall M \in \mathbb{Z}_N$.

We have $ed \equiv 1 \mod \varphi(N)$ and $\varphi(N) = (P-1)(Q-1)$, so $(P-1)(Q-1)|ed-1 \Rightarrow P-1|ed-1 \Rightarrow ed = \alpha(P-1)+1$ for some $\alpha \in \mathbb{Z}$
Thus $M^{ed} \equiv (M^{P-1})^\alpha \cdot M \equiv M \mod P$ by Fermat.
Analogously: $M^{ed} \equiv M \mod Q \Rightarrow M^{ed} \equiv M \mod N$

5.2 Semantic Security for Public Key Procedures

A public key procedure is semantically secure if for every M-distribution of messages of equal length, every function f and every PPT-algorithm A, there exists a PPT-algorithm B such that

$$Pr[A(1^k, pk, Enc(pk, M)) = f(M)] - Pr[B(1^k) = f(M)]$$

is negligibly small.

5.3 IND-CPA for Asymmetric Encryption

- Challenger C creates pair of keys $(pk, sk) \leftarrow Gen(1^k)$
- No Enc-oracle, instead the attacker obtains pk

5.4 Security of RSA

- Not semantically secure
  - $f(M) \equiv M^e \mod N$ can be calculated with ciphertext, but without ciphertext there's no information on M. This makes use of the determinism.
- Homomorphy
  - In $\mathbb{Z}_N$ we have $Enc(pk, M) \cdot Enc(pk, M') = M^e \cdot M'^e = (M \cdot M')^e = Enc(pk, M \cdot M')$.

5.5 RSA Padding

- Randomized padding
  - $pad(M, R) = M||0^l||R$, where $M, R \ll N$ and R random
  - $Enc(pk, M) = (pad(M, R))^e \mod N$
  - Dec gets and checks $pad(M, R)$ then extracts M
- RSA-OAEP contains pad-functionality (G, H are hashfunctions)
  - Heuristically as secure as inverting RSA-function
  - Best known attack: factorize N, so N of 2048 Bit is secure
  - $\ominus$ computationally intensive, hard to parallelize
  - $\oplus$ easy to implement

5.6 ElGamal

- Cyclic group $\mathbb{G} = \langle g \rangle, pk = (\mathbb{G}, g, g^x), sk = (\mathbb{G}, g, x)$ with x random
- $Enc(pk, M) = (g^y, g^{xy} \cdot M)$ with y random
- $Dec(sk, (Y, Z)) = Z/Y^x = (g^{xy} \cdot M)/(g^y)^x = M$
- Encryption is probabilistic, but also homomorph:

$$\begin{aligned} Enc(pk, M) \cdot Enc(pk, M') &= (g^y, g^{xy} \cdot M) \cdot (g^{y'}, g^{xy'} \cdot M') \\ &= (g^{y+y'}, g^{x(y+y')} \cdot M \cdot M') \\ &= Enc(pk, M \cdot M') \end{aligned}$$

- Semantically secure, non-homomorph variants exist
- Candidates for $\mathbb{G}$:
  - (real) subgroups of $\mathbb{Z}_p^*$, with p prime
  - subgroups of $\mathbb{F}_q^*$, with q a prime power
  - efficient: subgroup of elliptical curve $E(\mathbb{F}_q)$

6 Asymmetric Authentification of Messages

- Idea:
  - $(pk, sk) \leftarrow Gen(1^k)$ as with public key procedures
  - Signing: $\sigma \leftarrow Sig(sk, M)$
  - Verification: $Ver(pk, M, \sigma) \in \{0, 1\}$
  - Correctness as with MACs: $Ver(pk, M, \sigma) = 1 \; \forall(pk, sk) \leftarrow Gen(1^k), \forall M, \forall \sigma = Sig(sk, M)$

6.1 Security: EUF-CMA definition as with MACs

- Challenger C executes $(pk, sk) \leftarrow Gen(1^k)$ and provides A with a $Sig(sk, \cdot)$-oracle

6.2 RSA as a Signing Scheme

- $Sig(sk, M) \equiv M^d \mod N$
- $Ver(pk, M, \sigma) = 1 :\iff M \equiv \sigma^e \mod N$
- Problem: nonsense messages can be signed
  1. *First*, choose any $\sigma \in \mathbb{Z}_N$
  2. Let $M := \sigma^e \mod N$
  - Breaks EUF-CMA
- Problem: Homomorphy
  - Known signatures can be used to calculate new ones

6.3 RSA-PSS: "Probabilistic Signature Scheme"

- Preprocessing (Padding) of messages
- $Sig(sk, M) = (pad(M))^d \mod N$
- $Ver(pk, M, \sigma) = 1 :\iff \sigma^e \mod N$ is valid $pad(M)$
- Security of RSA-PSS: heuristic EUF-CMA-secure, if RSA-function is hard to invert

6.4 ElGamal Signatures

- Let $a := g^e$ for random e, b solution of $a \cdot x + e \cdot b \equiv M \mod |\mathbb{G}|$
- Then $Sig(sk, M) = (a, b)$
- $Ver(pk, M, \sigma) = 1 :\iff (g^x)^a a^b = g^M$