A Practice-Oriented Treatment of Pseudorandom Number Generators

Anand Desai (NTT MCL)
Alejandro Hevia (UC San Diego)
Yiqun Lisa Yin (NTT MCL)

Talk Outline

- Background
- Security definitions
- Analysis of the two most widely-used PRNGs (ANSI X9.17 and FIPS 186)
- Other security considerations

Background (1)

- Cryptographic PRNGs are required for virtually every cryptographic application.
- There exist provably-secure PRNGs under number-theoretic assumptions [BM84,BBS86,G00].
  - Not the most popular ones: efficiency is an issue.

Background (2)

- Most popular PRNGs use block ciphers or hash functions as the underlying primitive
- Standardized PRNGs
  - The ANSI X9.17 PRNG
  - The FIPS 186 PRNG

There have been no security proofs (under any reasonable assumption) that these PRNGs are secure.

Related Work

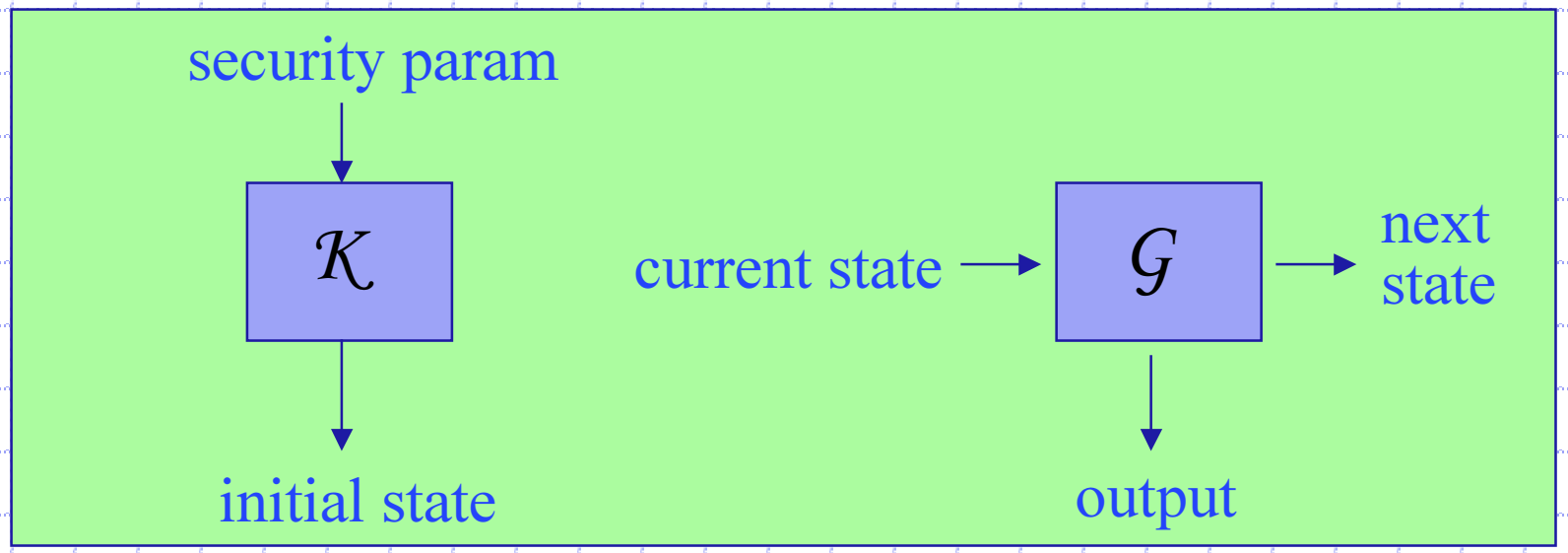
- There is extensive literature on the theory of PRNGs [Y82,BM84,BBS86,HILL89].
- Results on block-cipher-based PRNGs focus on provably-secure design [ARV99] and generic forward security techniques [BY01,AB00].
- Previous analyses [KSWH98,G98,B01] identified weaknesses but were mostly ad-hoc.

Our Contributions

- Analysis framework more suitable for PRNGs as used in practice
- Analysis of the ANSI X9.17 and FIPS 186 PRNGs
  - Formalize assumptions on primitives
  - Suggest guidelines on secure usage
  - Identify improvements

PRNGs in cryptography

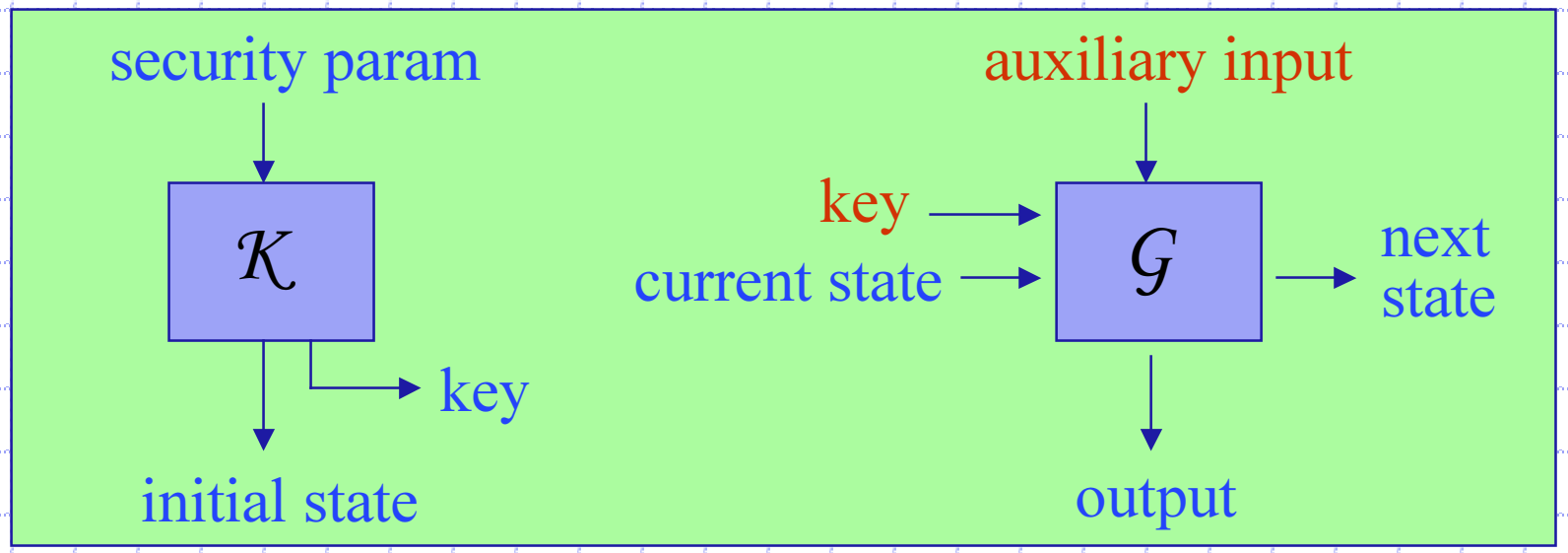
A PRNG $\mathcal{GE} = (\mathcal{K}, \mathcal{G})$ is a pair of stateful algorithms



$\mathcal{G} : current\ state \rightarrow next\ state \times output$

PRNGs as used in practice

PRNGs are extended so $\mathcal{G}$ takes additional inputs



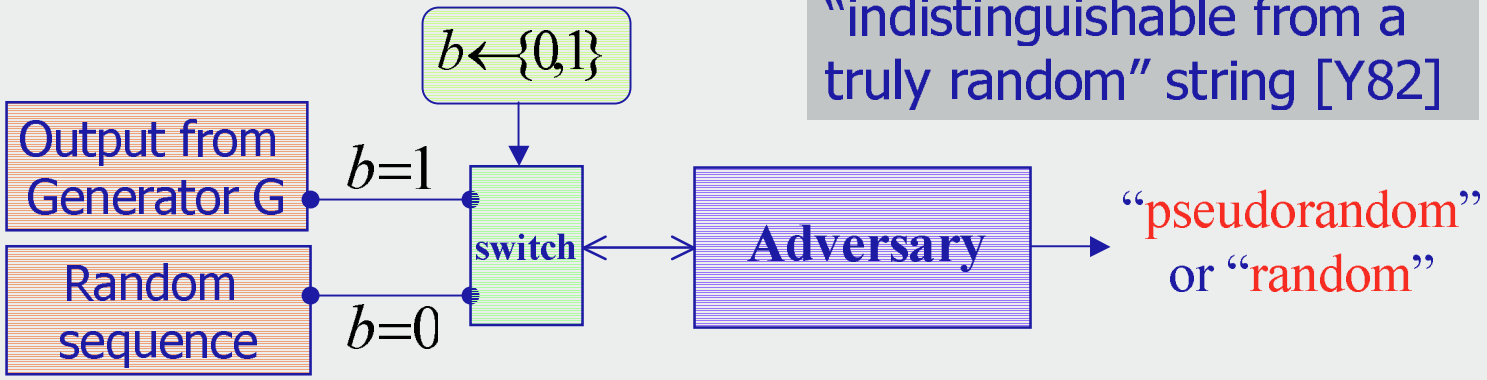
$$\mathcal{G} : key \times current\ state \times auxiliary\ input \rightarrow next\ state \times output$$

PRNGs: Theory vs. Practice

PRNGs used in cryptography	PRNGs used in practice
States are assumed hidden at all times	◆ Take "auxiliary inputs" (e.g. timestamps) ◆ May leak out current state over time ◆ Are based on secret-key or keyless primitives

Towards a security definition (1)

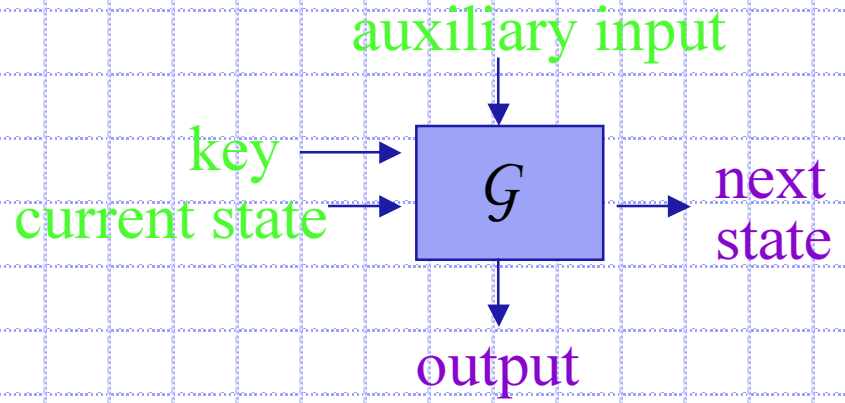
Security goal



The output should be "indistinguishable from a truly random" string [Y82]

Attacker Capabilities

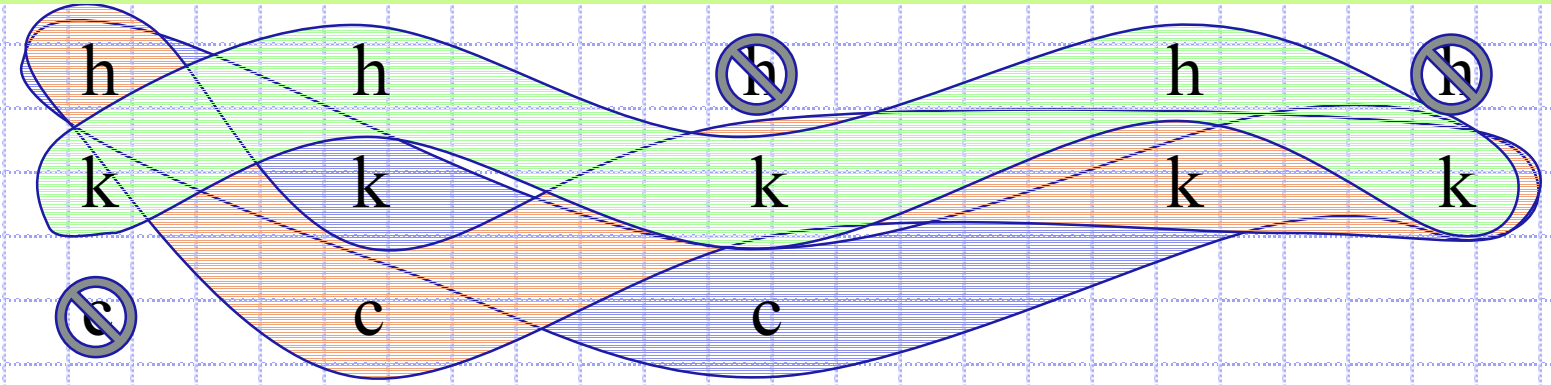
Inputs	Outputs
hidden known chosen	hidden known



Towards a security definition (2)

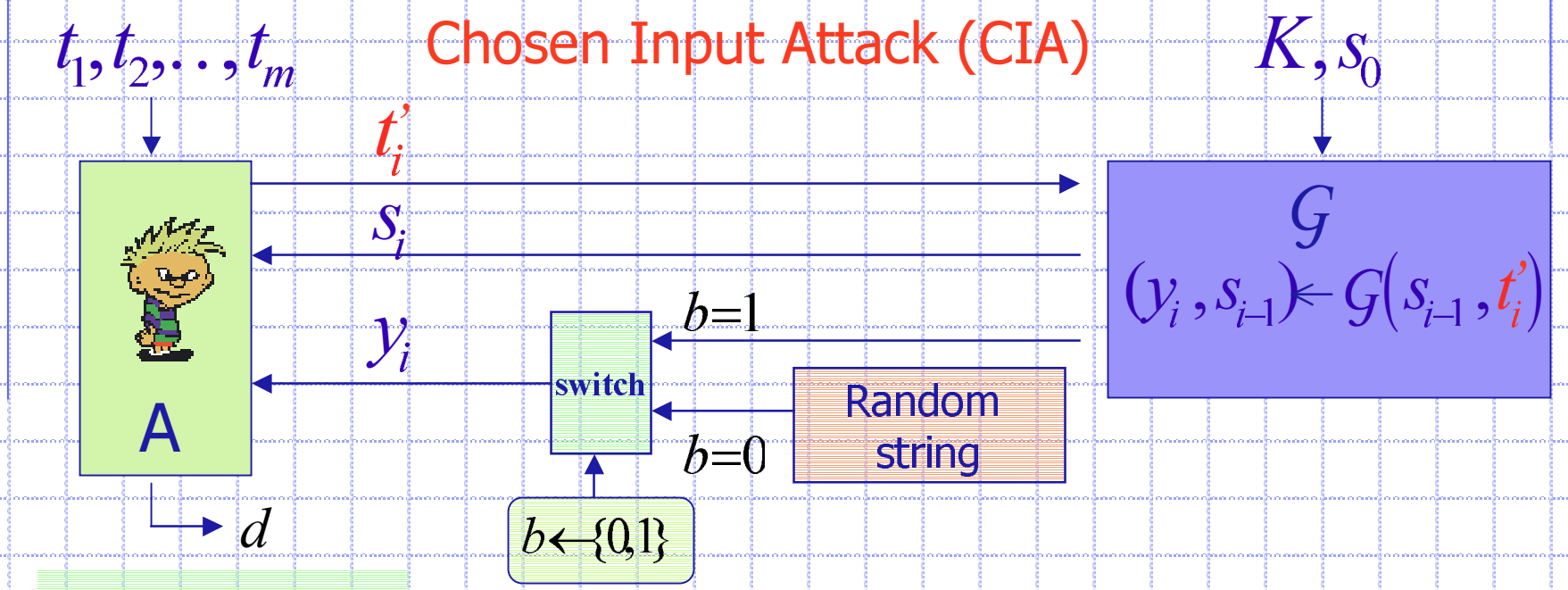
Attacker Viewpoint

$$G : key \times current\ state \times auxiliary\ input \rightarrow next\ state \times output$$



Attack Name	key	current state	aux input	next state
Chosen Input Attack	hidden	known	chosen	known
Chosen State Attack	hidden	chosen	known	known
Known Key Attack	known	hidden	known	hidden

Security of PRNGs (1)

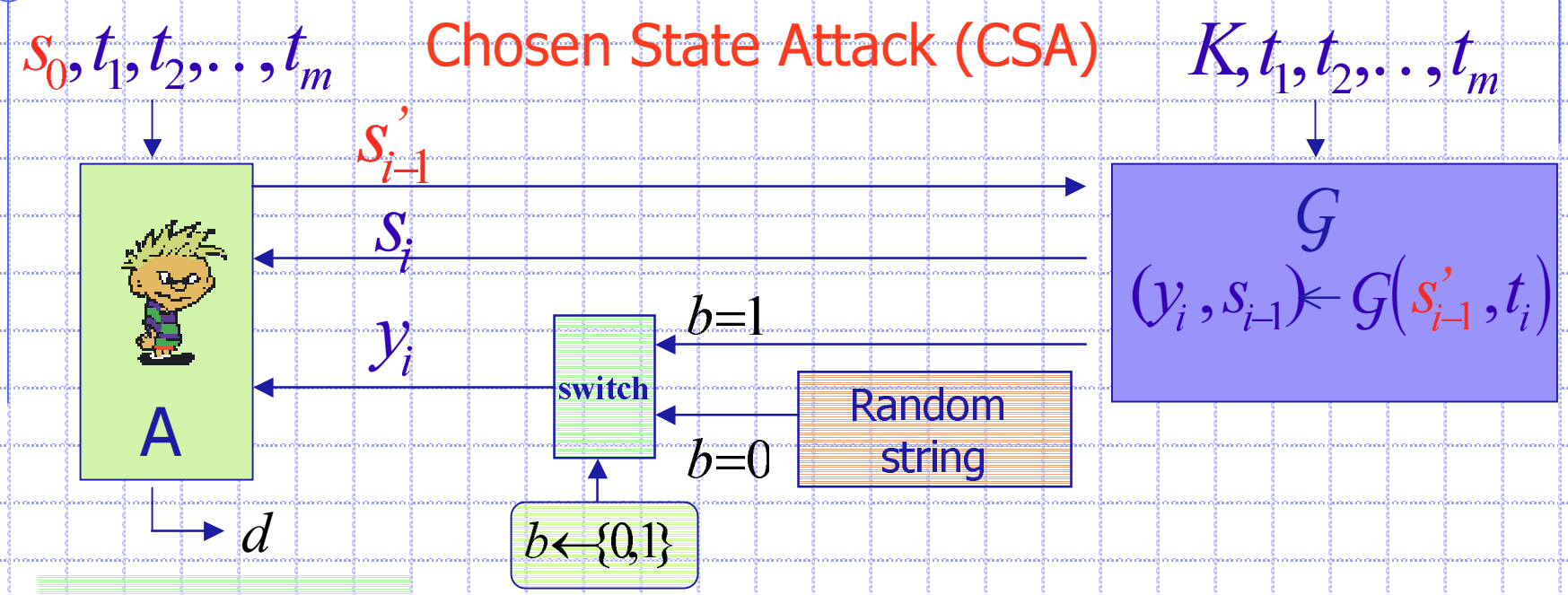


A wins if $d=b$

$$\mathrm{Adv}^{\text{prg-cia}}_{\mathcal{GE},m}(t) = \max_A \{2\Pr[A \text{ wins}] - 1\}$$

We want $\mathrm{Adv}^{\text{prg-cia}}_{\mathcal{GE},m}(t)$ "small" for "large" t

Security of PRNGs (2)

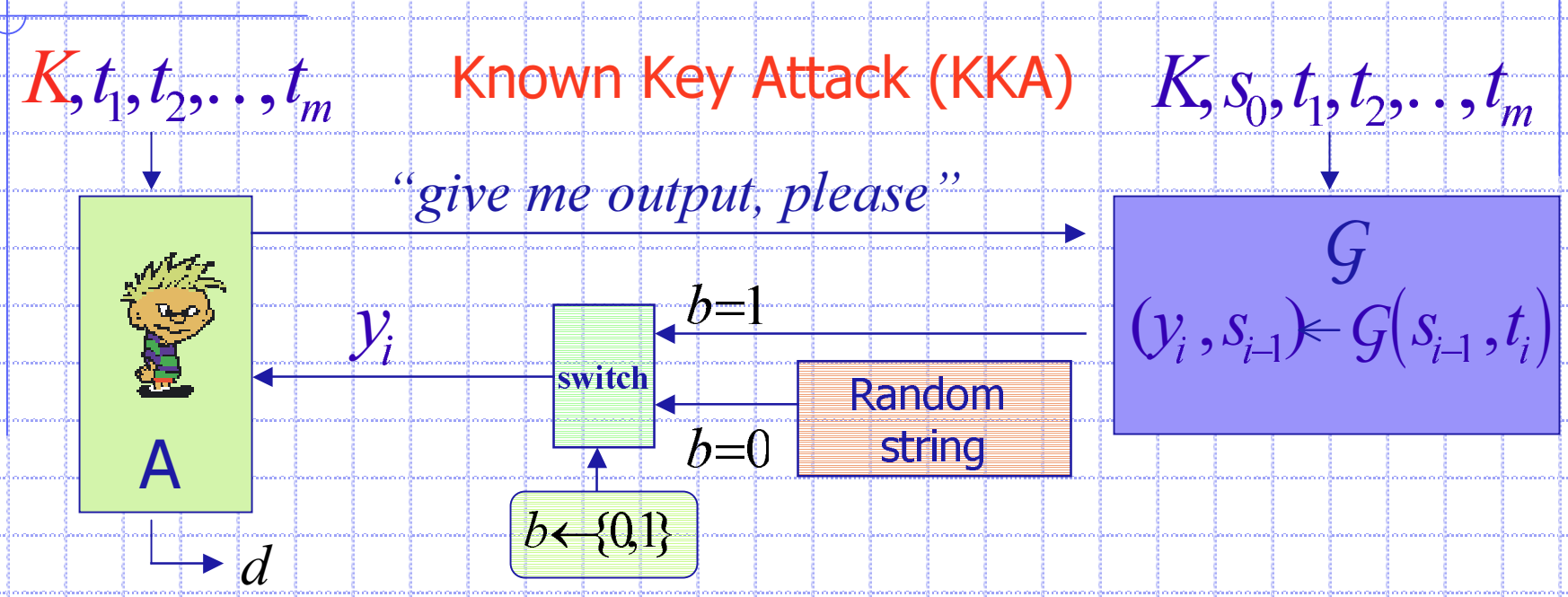


A wins if $d=b$

$$\mathrm{Adv}^{\text{prg-csa}}_{\mathcal{GE},m}(t) = \max_A \{2\Pr[A \text{ wins}] - 1\}$$

We want $\mathrm{Adv}^{\text{prg-csa}}_{\mathcal{GE},m}(t)$ "small" for "large" t

Security of PRNGs (3)

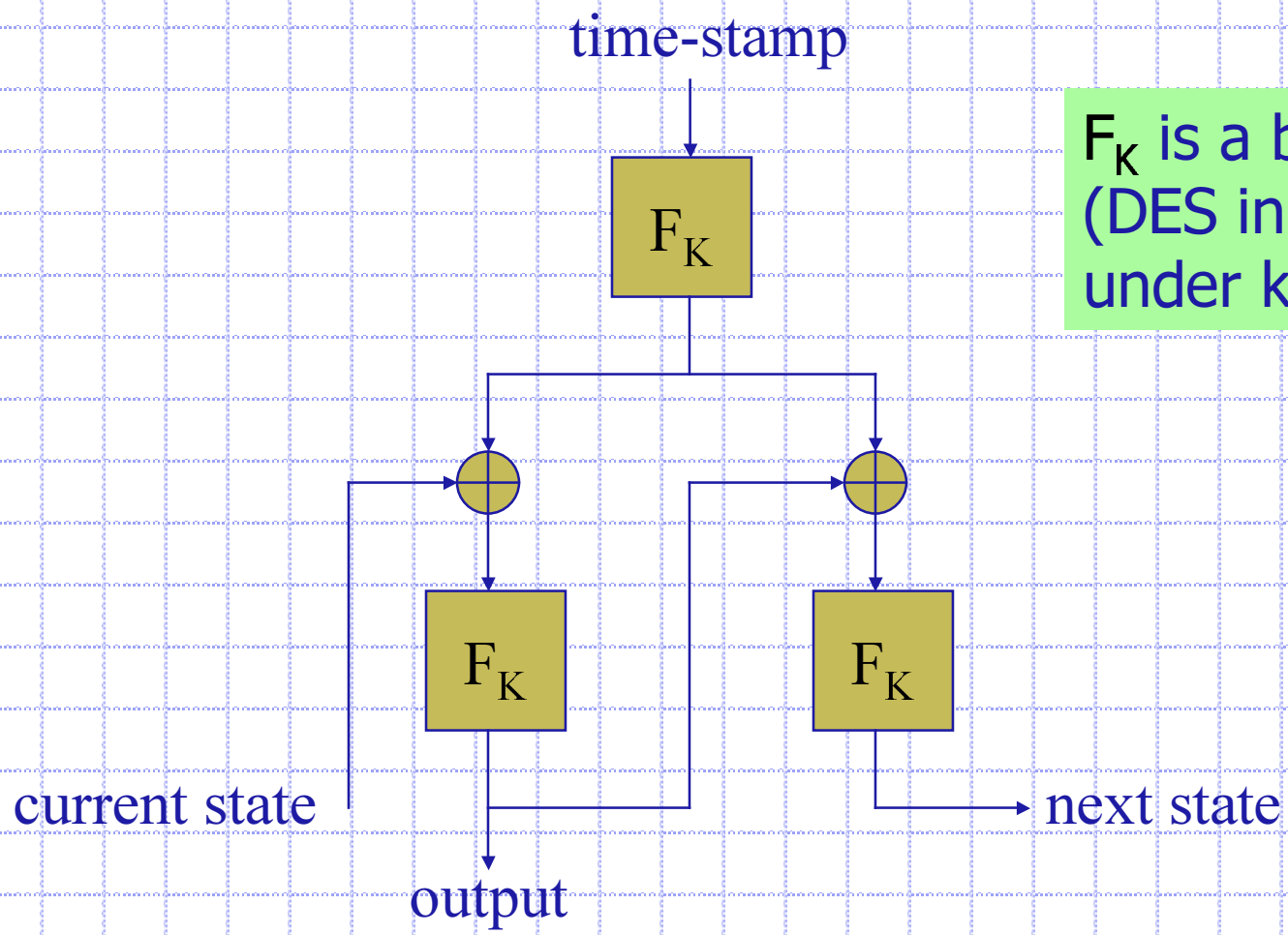


A wins if $d=b$

$$\mathrm{Adv}^{\text{prg-kka}}_{G\mathcal{E},m}(t) = \max_A \{2\Pr[A \text{ wins}] - 1\}$$

We want $\mathrm{Adv}^{\text{prg-kka}}_{G\mathcal{E},m}(t)$ "small" for "large" t

The ANSI X9.17 PRNG



F_K is a block cipher (DES in the spec) under key K

ANSI PRNG: Security results (1)

- Insecure under any attack if key is known.
- Insecure under an attack where both the input and current state may be chosen.

ANSI PRNG: Security results (2)

ANSI PRNG is secure under Chosen Input Attack and Chosen State Attack assuming the underlying block cipher is a pseudorandom permutation (PRP).

Theorem: Let $\mathcal{GE}$ be the ANSI X9.17 PRNG based on a function family F. Then

$$Adv^{prg-csa}_{\mathcal{GE},m}(t) \leq 2 \cdot Adv^{prp}_{F}(t,3m) + m \cdot (13m-2) \cdot 2^{-n-1}$$

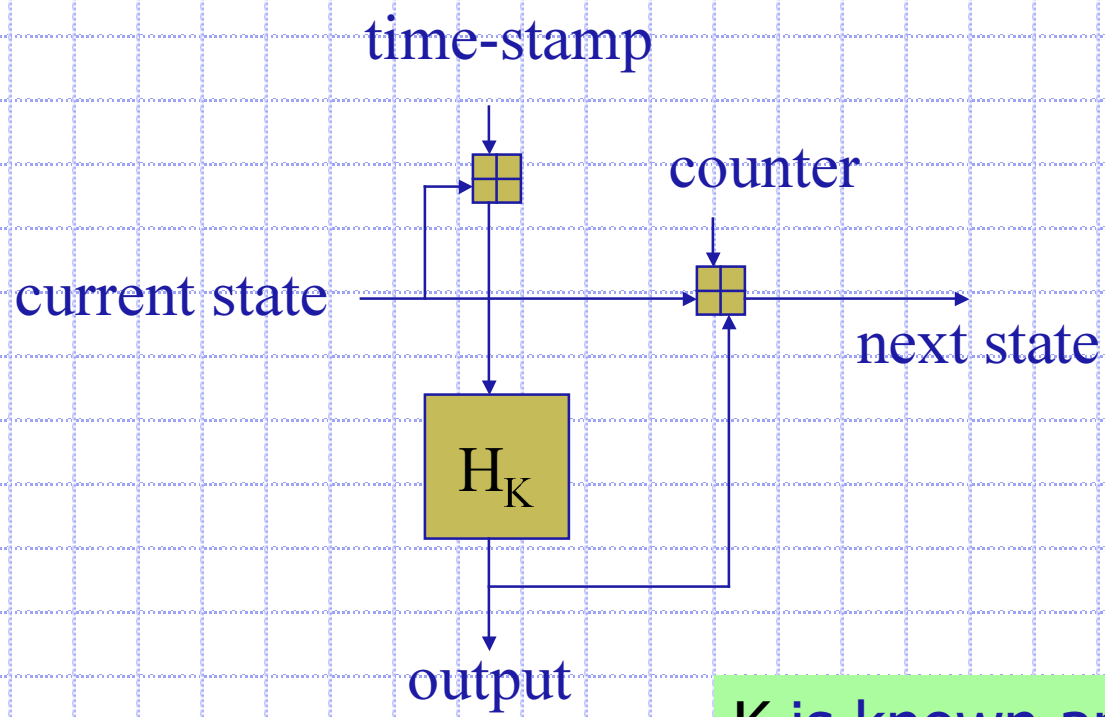
$$Adv^{prg-cia}_{\mathcal{GE},m}(t) \leq 2 \cdot Adv^{prp}_{F}(t,3m) + \left((4m-1)^2 + m^2 + 1\right) \cdot 2^{-n-1}$$

where m is the number of n-bit output blocks.

ANSI PRNG: Remarks

- Throughput can be doubled by outputting intermediate states as part of PRNG output
  - Secrecy of intermediate states is unnecessary
  - Intermediate states are pseudorandom
- "Good" randomness is better used on key (rather than on state)

The FIPS 186 PRNG



K is known and fixed in FIPS 186 specification

FIPS PRNG: Attacks

- Insecure under any attack where state is known.
- Insecure under any attack if the input may be chosen [KSWH98].

FIPS PRNG: Towards an Analysis

We need reasonable assumptions on H

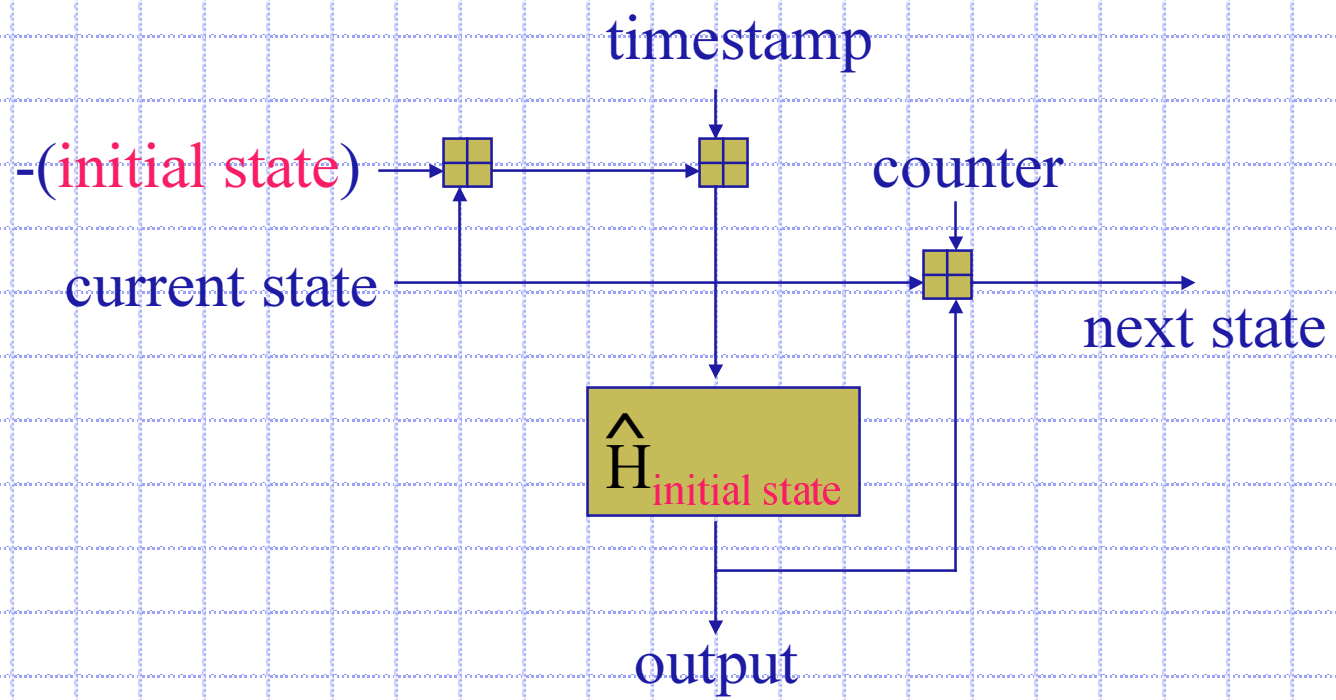
- Collision Resistance? does not suffice
- Random Oracle? overkill

$\hat{H}_s(x) = H_K(s+x)$ can be seen as secret-key hash function if s is secret

→ Assume $\hat{H}_s$ is a PRF family

- Similar assumptions have been made before [BGR95,BCK96a,BCK96b,ARV99].
- No known attacks seem to contradict this assumption.

FIPS PRNG: An alternative view



Different looking implementation but same input/output characteristics

FIPS PRNG: Security results

FIPS PRNG is secure under Known Key Attack assuming the underlying primitive (in the alternative view) is a PRF.

Theorem: Let $\mathcal{GE}$ be the FIPS 186 PRNG based on the function family $\hat{H}$. Then

$$Adv^{prg-kka}_{\mathcal{GE},m}(t) \leq 2 \cdot Adv^{prf}_{\hat{H}}(t,m) + m \cdot (m-1) \cdot 2^{-n-1}$$

where m is the number of n-bit output blocks.

Other Considerations

- Most other PRNGs used in practice bear similarities with the two PRNGs analyzed
- Preserving security even under a break-in (Forward Security) seems desirable…

  But neither the ANSI nor the FIPS PRNG are forward-secure.

Conclusions

- We propose a framework more suitable for PRNGs as used in practice
- ANSI X9.17 PRNG
  - Secure if either state or inputs are not chosen
  - Randomness is better used in key
  - Throughput can be doubled by outputting state
- FIPS 186 PRNGs
  - Secure if states are hidden and inputs are not chosen
- For both, we formalize assumptions needed on primitives.