The World of Crypto Primitives

RSA Assumption

TDP

DDH Assumption

CDH Assumption

Factoring Assumption

CPA-Secure PKE

Key Agreement

DL Assumption

CRHF

OWF

PRG

PRF\PRP

MAC

Signatures

Commitments

CPA\CCA-Secure Symmetric-Key Encryption

ZK Proofs for NP
Assumptions, Primitives & Protocols

- RSA Assumption
- TDP
- DDH Assumption
- CPA-Secure PKE
- CDH Assumption
- DL Assumption
- CRHF

Protocols

- CRHF
- DL
- Factoring Assumption

- CPA-Secure PKE
- Key Agreement

- CPA/CCA-Secure Encryption

- OWF
- PRG
- PRF/PRP
- MAC
- Signatures
- Commitments

- Symmetric-Key Encryption
- ZK Proofs for NP
Recall: Modern Cryptography

The scientific study of techniques for designing systems that withstand adversarial behavior

- What does “secure” mean?
- Can we avoid “break → repair → break → repair → …”?
- Can we prove “security”?
Recall: Modern Cryptography

Analyzing the security of a system consists of
1. Formalizing a precise definition of security
2. Stating the underlying assumptions (if any)
3. Proving that the definition is satisfied given the assumptions

Security definition =
computational ability × type of attack × notion of “break”
When Designing a Cryptographic System

You should always ask yourself

1. What is the functionality of the system?
2. How do I expect the system to get attacked? Can I formalize a security definition?

Security definition =
computational ability × type of attack × notion of “break”
When Designing a Cryptographic System

You should always ask yourself

1. What is the functionality of the system?
2. How do I expect the system to get attacked? Can I formalize a security definition?
3. What are my building blocks? Are they used in an intuitive way?
4. What assumptions am I making about the security of the building blocks?
5. Can I reduce the security of the system to that of the building blocks?
Cryptographic systems still get broken

WHY?
Assumption X turns out false
- Somewhat unlikely (in the near future) for DDH, DL, RSA,…
- Avoid inventing your own assumptions

CPA-secure PKE turns out insecure
- Somewhat unlikely for El-Gamal, RSA,…
- Avoid inventing your own basic primitives
How can this get broken?

**Bad implementation**
- Use well-known & publicly-reviewed crypto libraries
- Avoid writing your own code unless necessary

**Attacks outside the model**
- Does CPA security suffice for System Y? What about CCA security?
- Is the adversarial model for System Y sufficiently realistic?
- Side-channel attacks (key leakage, timing, noise, ...)

Assumption $X$ → CPA-secure PKE → System Y
How can this get broken?

Weakest point of failure

- Uneducated users
- Insider threats, malware,...
- Weak keys (OS supplied randomness?)
- ??
Modern cryptography provides us with extremely powerful approaches & tools

Should be rigorously used while understanding the possible limitations