Project Ideas

Project Ideas

Lecture 13

Categories

Categories

- Implementations

Categories

- Implementations
- Formalizing security of applied-crypto works

Categories

- Implementations
- Formalizing security of applied-crypto works
- Conceiving new applications

Categories

- Implementations
- Formalizing security of applied-crypto works
- Conceiving new applications
- Surveys

Categories

- Implementations
- Formalizing security of applied-crypto works
- Conceiving new applications
- Surveys
- Theoretical research, cryptanalysis, ...

Implementations

Implementations

- Security first: don't worry about efficiency

Implementations

- Security first: don't worry about efficiency
- Build from scratch: no legacy concerns

Implementations

- Security first: don't worry about efficiency
- Build from scratch: no legacy concerns
- Base security on simple intractability assumptions

Implementations

- Security first: don't worry about efficiency
- Build from scratch: no legacy concerns
- Base security on simple intractability assumptions
- Will have a security parameter

Implementations

- Security first: don't worry about efficiency
- Build from scratch: no legacy concerns
- Base security on simple intractability assumptions
- Will have a security parameter
- Modular: plug-and-play, with interfaces conforming to standard abstract notions (OWP, PRG, PRF, ...)

Implementations

- Security first: don't worry about efficiency
- Build from scratch: no legacy concerns
- Base security on simple intractability assumptions
- Will have a security parameter
- Modular: plug-and-play, with interfaces conforming to standard abstract notions (OWP, PRG, PRF, ...)
- IVORY-CRYPT

Implementations

- Security first: don't worry about efficiency
- Build from scratch: no legacy concerns
- Base security on simple intractability assumptions
- Will have a security parameter
- Modular: plug-and-play, with interfaces conforming to standard abstract notions (OWP, PRG, PRF, ...)
- IVORY-CRYPT
  - Cryptography from the ivory towers :-)

Ivory-Crypt

Ivory-Crypt

- PRF

Ivory-Crypt

- PRF
  - Based on arbitrary PRG [Goldreich-Goldwasser-Micali]

Ivory-Crypt

- PRF
  - Based on arbitrary PRG [Goldreich-Goldwasser-Micali]
    - In turn based on arbitrary OWP [Yao,Goldreich-Levin]

Ivory-Crypt

- PRF
  - Based on arbitrary PRG [Goldreich-Goldwasser-Micali]
    - In turn based on arbitrary OWP [Yao,Goldreich-Levin]
  - Or based on DDH assumption [Naor-Reingold]

Ivory-Crypt

- PRF
  - Based on arbitrary PRG [Goldreich-Goldwasser-Micali]
    - In turn based on arbitrary OWP [Yao,Goldreich-Levin]
  - Or based on DDH assumption [Naor-Reingold]
- Use it for shared-key secure communication channels

Ivory-Crypt

- PRF
  - Based on arbitrary PRG [Goldreich-Goldwasser-Micali]
    - In turn based on arbitrary OWP [Yao,Goldreich-Levin]
  - Or based on DDH assumption [Naor-Reingold]
- Use it for shared-key secure communication channels
  - First, implement authenticated communication streams

Ivory-Crypt

- PRF
  - Based on arbitrary PRG [Goldreich-Goldwasser-Micali]
    - In turn based on arbitrary OWP [Yao,Goldreich-Levin]
  - Or based on DDH assumption [Naor-Reingold]
- Use it for shared-key secure communication channels
  - First, implement authenticated communication streams
  - Implement (CPA secure) encryption for each stream

Ivory-Crypt

- PRF
  - Based on arbitrary PRG [Goldreich-Goldwasser-Micali]
    - In turn based on arbitrary OWP [Yao,Goldreich-Levin]
  - Or based on DDH assumption [Naor-Reingold]
- Use it for shared-key secure communication channels
  - First, implement authenticated communication streams
  - Implement (CPA secure) encryption for each stream
- Sketch formal security guarantees and proofs

Ivory-Crypt

Ivory-Crypt

- Public-Key Secure communication

Ivory-Crypt

- Public-Key Secure communication
  - Most useful setting: Only server has public-key

Ivory-Crypt

- Public-Key Secure communication
  - Most useful setting: Only server has public-key
  - Design/analyze and build all components

Ivory-Crypt

- Public-Key Secure communication
  - Most useful setting: Only server has public-key
  - Design/analyze and build all components
    - Digital signatures and authenticated channel (using say OWF, and Discrete Log-based CRHF, or OWP-based UOWHF)

Ivory-Crypt

- Public-Key Secure communication
  - Most useful setting: Only server has public-key
  - Design/analyze and build all components
    - Digital signatures and authenticated channel (using say OWF, and Discrete Log-based CRHF, or OWP-based UOWHF)
    - Key-agreement/PKE

Ivory-Crypt

- Public-Key Secure communication
  - Most useful setting: Only server has public-key
  - Design/analyze and build all components
    - Digital signatures and authenticated channel (using say OWF, and Discrete Log-based CRHF, or OWP-based UOWHF)
    - Key-agreement/PKE
    - Secure communication given a shared key and authenticated channel (may use a shared-key secure communication channel module: could be implemented as a separate project)

Ivory-Crypt

- Public-Key Secure communication
  - Most useful setting: Only server has public-key
  - Design/analyze and build all components
    - Digital signatures and authenticated channel (using say OWF, and Discrete Log-based CRHF, or OWP-based UOWHF)
    - Key-agreement/PKE
    - Secure communication given a shared key and authenticated channel (may use a shared-key secure communication channel module: could be implemented as a separate project)
- SSH: a candidate app for not-so-efficient crypto

Ivory-Crypt

Ivory-Crypt

- An Achilles Heel

Ivory-Crypt

- An Achilles Heel
  - Source of randomness

Ivory-Crypt

- An Achilles Heel
  - Source of randomness
- Randomness extractors

Ivory-Crypt

- An Achilles Heel
  - Source of randomness
- Randomness extractors
  - Purify a "weak" random source (or multiple sources)

Ivory-Crypt

- An Achilles Heel
  - Source of randomness
- Randomness extractors
  - Purify a "weak" random source (or multiple sources)
  - May need a small amount of "seed randomness" as catalyst

Ivory-Crypt

- An Achilles Heel
  - Source of randomness
- Randomness extractors
  - Purify a "weak" random source (or multiple sources)
  - May need a small amount of "seed randomness" as catalyst
  - Implement different ones from the literature and compare performance for reasonable parameters

Ivory-Crypt

- An Achilles Heel
  - Source of randomness
- Randomness extractors
  - Purify a "weak" random source (or multiple sources)
  - May need a small amount of "seed randomness" as catalyst
  - Implement different ones from the literature and compare performance for reasonable parameters
- (Another Achilles Heel: Side-channels. Handled by "Leakage-resilient cryptography")

Ivory-Crypt

Ivory-Crypt

- Secure Multi-Party Computation (SMC, or MPC)

Ivory-Crypt

- Secure Multi-Party Computation (SMC, or MPC)
  - (Will see next week, and later)

Ivory-Crypt

- Secure Multi-Party Computation (SMC, or MPC)
  - (Will see next week, and later)
  - Honest-but-curious setting: a few implementations already out there (as also, an "in-house" implementation)

Ivory-Crypt

- Secure Multi-Party Computation (SMC, or MPC)
  - (Will see next week, and later)
  - Honest-but-curious setting: a few implementations already out there (as also, an "in-house" implementation)
    - Add to them/Enhance them

Ivory-Crypt

- Secure Multi-Party Computation (SMC, or MPC)
  - (Will see next week, and later)
  - Honest-but-curious setting: a few implementations already out there (as also, an "in-house" implementation)
    - Add to them/Enhance them
  - Honest-majority setting: No "crypto" -- i.e., information theoretic security (except secure communication channels)

Ivory-Crypt

- Secure Multi-Party Computation (SMC, or MPC)
  - (Will see next week, and later)
  - Honest-but-curious setting: a few implementations already out there (as also, an "in-house" implementation)
    - Add to them/Enhance them
  - Honest-majority setting: No "crypto" -- i.e., information theoretic security (except secure communication channels)
  - No honesty setting. (Will take too long to implement?)

Ivory-Crypt

- Secure Multi-Party Computation (SMC, or MPC)
  - (Will see next week, and later)
  - Honest-but-curious setting: a few implementations already out there (as also, an "in-house" implementation)
    - Add to them/Enhance them
  - Honest-majority setting: No "crypto" -- i.e., information theoretic security (except secure communication channels)
  - No honesty setting. (Will take too long to implement?)
    - Stand-alone, and Composable

New Applications

New Applications

- Applying crypto tools

New Applications

- Applying crypto tools
  - Conceive and design (on paper) new systems which can use tools that we will see in the remainder of the course

New Applications

- Applying crypto tools
  - Conceive and design (on paper) new systems which can use tools that we will see in the remainder of the course
    - e.g. Homomorphic encryption schemes

New Applications

- Applying crypto tools
  - Conceive and design (on paper) new systems which can use tools that we will see in the remainder of the course
    - e.g. Homomorphic encryption schemes
      - Non-malleable homomorphic encryption schemes?

New Applications

- Applying crypto tools
  - Conceive and design (on paper) new systems which can use tools that we will see in the remainder of the course
    - e.g. Homomorphic encryption schemes
      - Non-malleable homomorphic encryption schemes?
    - e.g. SMC concepts, Private Information Retrieval (PIR)

New Applications

- Applying crypto tools
  - Conceive and design (on paper) new systems which can use tools that we will see in the remainder of the course
    - e.g. Homomorphic encryption schemes
      - Non-malleable homomorphic encryption schemes?
    - e.g. SMC concepts, Private Information Retrieval (PIR)
  - For e.g., "distributed secure storage" using SMC or PIR ideas

New Applications

- Applying crypto tools
  - Conceive and design (on paper) new systems which can use tools that we will see in the remainder of the course
    - e.g. Homomorphic encryption schemes
      - Non-malleable homomorphic encryption schemes?
    - e.g. SMC concepts, Private Information Retrieval (PIR)
  - For e.g., "distributed secure storage" using SMC or PIR ideas
- Must give a security analysis (definition and proof)