An Intrusion Tolerant Threshold Cryptographic System

Kamran Riaz Khan
<krkhan@inspirated.com>

March 2, 2010

Outline

Symmetric-Key Cryptography

- Alice calculates:

$$c := E(K, m) \tag{1}$$

- Alice sends c to Bob
- Bob calculates:

$$m := D(K, c) \tag{2}$$

- How to communicate K?

Symmetric-Key Cryptography

- Alice calculates:

$$c := E(K, m) \tag{1}$$

- Alice sends c to Bob
- Bob calculates:

$$m := D(K, c) \tag{2}$$

- How to communicate K?

Symmetric-Key Cryptography

- Alice calculates:

$$c := E(K, m) \tag{1}$$

- Alice sends c to Bob
- Bob calculates:

$$m := D(K, c) \tag{2}$$

- How to communicate K?

Symmetric-Key Cryptography

- Alice calculates:

$$c := E(K, m) \tag{1}$$

- Alice sends c to Bob
- Bob calculates:

$$m := D(K, c) \tag{2}$$

- How to communicate K?

Symmetric-Key Cryptography

- Alice calculates:

$$c := E(K, m) \tag{1}$$

- Alice sends c to Bob
- Bob calculates:

$$m := D(K, c) \tag{2}$$

- How to communicate K?

Public-Key Cryptography

- Bank generates a pair of keys (S_{bank}, P_{bank}) such that

$$D(S_{bank}, E(P_{bank}, m)) = m \quad (3)$$

  for all values of m
- P_{bank} is published

Public-Key Cryptography

- Bank generates a pair of keys (S_{bank}, P_{bank}) such that

$$D(S_{bank}, E(P_{bank}, m)) = m \tag{3}$$

for all values of m

- P_{bank} is published

Public-Key Cryptography

- Bank generates a pair of keys (S_{bank}, P_{bank}) such that

$$D(S_{bank}, E(P_{bank}, m)) = m \quad (3)$$

  for all values of m
- P_{bank} is published

Public-Key Cryptography

- For credit card number m, client calculates:

$$c := E(P_{bank}, m) \tag{4}$$

- Client sends c to bank
- Bank receives c and calculates:

$$m := D(S_{bank}, c) \tag{5}$$

- Equation (3) ensures m is recovered from c *

*N. Ferguson and B. Schneier, *Practical Cryptography*. New York, NY, USA: John Wiley & Sons, Inc., 2003 [1]

Public-Key Cryptography

- For credit card number m, client calculates:

$$c := E(P_{bank}, m) \tag{4}$$

- Client sends c to bank
- Bank receives c and calculates:

$$m := D(S_{bank}, c) \tag{5}$$

- Equation (3) ensures m is recovered from c *

*N. Ferguson and B. Schneier, *Practical Cryptography*. New York, NY, USA: John Wiley & Sons, Inc., 2003 [1]

Public-Key Cryptography

- For credit card number m, client calculates:

$$c := E(P_{bank}, m) \tag{4}$$

- Client sends c to bank
- Bank receives c and calculates:

$$m := D(S_{bank}, c) \tag{5}$$

- Equation (3) ensures m is recovered from c *

---

*N. Ferguson and B. Schneier, *Practical Cryptography*. New York, NY, USA: John Wiley & Sons, Inc., 2003 [1]

Public-Key Cryptography

- For credit card number m, client calculates:

$$c := E(P_{bank}, m) \tag{4}$$

- Client sends c to bank
- Bank receives c and calculates:

$$m := D(S_{bank}, c) \tag{5}$$

- Equation (3) ensures m is recovered from c *

*N. Ferguson and B. Schneier, *Practical Cryptography*. New York, NY, USA: John Wiley & Sons, Inc., 2003 [1]

Public-Key Cryptography

- For credit card number m, client calculates:

$$c := E(P_{bank}, m) \tag{4}$$

- Client sends c to bank
- Bank receives c and calculates:

$$m := D(S_{bank}, c) \tag{5}$$

- Equation (3) ensures m is recovered from c *

*N. Ferguson and B. Schneier, *Practical Cryptography*. New York, NY, USA: John Wiley & Sons, Inc., 2003 [1]

Postal Analogy

Symmetric-Key:

- Alice buys $\text{Padlock}_{\text{Alice}}$ and $\text{Key}_{\text{Alice}}$
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Alice}}$
- Alice sends the box to Bob
- Alice gives Bob the $\text{Key}_{\text{Alice}}$ through some other channel
- Bob receives the box
- Bob unlocks the box using $\text{Key}_{\text{Alice}}$

Postal Analogy

Symmetric-Key:

- Alice buys $\text{Padlock}_{\text{Alice}}$ and $\text{Key}_{\text{Alice}}$
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Alice}}$
- Alice sends the box to Bob
- Alice gives Bob the $\text{Key}_{\text{Alice}}$ through some other channel
- Bob receives the box
- Bob unlocks the box using $\text{Key}_{\text{Alice}}$

Postal Analogy

Symmetric-Key:

- Alice buys $\text{Padlock}_{\text{Alice}}$ and $\text{Key}_{\text{Alice}}$
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Alice}}$
- Alice sends the box to Bob
- Alice gives Bob the $\text{Key}_{\text{Alice}}$ through some other channel
- Bob receives the box
- Bob unlocks the box using $\text{Key}_{\text{Alice}}$

Postal Analogy

Symmetric-Key:

- Alice buys $\text{Padlock}_{\text{Alice}}$ and $\text{Key}_{\text{Alice}}$
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Alice}}$
- Alice sends the box to Bob
- Alice gives Bob the $\text{Key}_{\text{Alice}}$ through some other channel
- Bob receives the box
- Bob unlocks the box using $\text{Key}_{\text{Alice}}$

Postal Analogy

Symmetric-Key:

- Alice buys $\text{Padlock}_{\text{Alice}}$ and $\text{Key}_{\text{Alice}}$
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Alice}}$
- Alice sends the box to Bob
- Alice gives Bob the $\text{Key}_{\text{Alice}}$ through some other channel
- Bob receives the box
- Bob unlocks the box using $\text{Key}_{\text{Alice}}$

Postal Analogy

Symmetric-Key:

- Alice buys $\text{Padlock}_{\text{Alice}}$ and $\text{Key}_{\text{Alice}}$
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Alice}}$
- Alice sends the box to Bob
- Alice gives Bob the $\text{Key}_{\text{Alice}}$ through some other channel
- Bob receives the box
- Bob unlocks the box using $\text{Key}_{\text{Alice}}$

Postal Analogy

Symmetric-Key:

- Alice buys $\text{Padlock}_{\text{Alice}}$ and $\text{Key}_{\text{Alice}}$
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Alice}}$
- Alice sends the box to Bob
- Alice gives Bob the $\text{Key}_{\text{Alice}}$ through some other channel
- Bob receives the box
- Bob unlocks the box using $\text{Key}_{\text{Alice}}$

Postal Analogy

Symmetric-Key:

- Alice buys $\text{Padlock}_{\text{Alice}}$ and $\text{Key}_{\text{Alice}}$
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Alice}}$
- Alice sends the box to Bob
- Alice gives Bob the $\text{Key}_{\text{Alice}}$ through some other channel
- Bob receives the box
- Bob unlocks the box using $\text{Key}_{\text{Alice}}$

Postal Analogy

Asymmetric-Key:

- Bob buys $\text{Padlock}_{\text{Bob}}$ and Key_{Bob}
- Bob sends $\text{Padlock}_{\text{Bob}}$ to Alice
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Bob}}$
- Alice sends the box to Bob
- Bob receives the box
- Bob unlocks the box using Key_{Bob}

Postal Analogy

Asymmetric-Key:

- Bob buys $\text{Padlock}_{\text{Bob}}$ and Key_{Bob}
- Bob sends $\text{Padlock}_{\text{Bob}}$ to Alice
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Bob}}$
- Alice sends the box to Bob
- Bob receives the box
- Bob unlocks the box using Key_{Bob}

Postal Analogy

Asymmetric-Key:

- Bob buys $\text{Padlock}_{\text{Bob}}$ and Key_{Bob}
- Bob sends $\text{Padlock}_{\text{Bob}}$ to Alice
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Bob}}$
- Alice sends the box to Bob
- Bob receives the box
- Bob unlocks the box using Key_{Bob}

Postal Analogy

Asymmetric-Key:

- Bob buys $\text{Padlock}_{\text{Bob}}$ and Key_{Bob}
- Bob sends $\text{Padlock}_{\text{Bob}}$ to Alice
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Bob}}$
- Alice sends the box to Bob
- Bob receives the box
- Bob unlocks the box using Key_{Bob}

Postal Analogy

Asymmetric-Key:

- Bob buys $\text{Padlock}_{\text{Bob}}$ and Key_{Bob}
- Bob sends $\text{Padlock}_{\text{Bob}}$ to Alice
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Bob}}$
- Alice sends the box to Bob
- Bob receives the box
- Bob unlocks the box using Key_{Bob}

Postal Analogy

Asymmetric-Key:

- Bob buys $\text{Padlock}_{\text{Bob}}$ and Key_{Bob}
- Bob sends $\text{Padlock}_{\text{Bob}}$ to Alice
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Bob}}$
- Alice sends the box to Bob
- Bob receives the box
- Bob unlocks the box using Key_{Bob}

Postal Analogy

Asymmetric-Key:

- Bob buys $\text{Padlock}_{\text{Bob}}$ and Key_{Bob}
- Bob sends $\text{Padlock}_{\text{Bob}}$ to Alice
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Bob}}$
- Alice sends the box to Bob
- Bob receives the box
- Bob unlocks the box using Key_{Bob}

Postal Analogy

Asymmetric-Key:

- Bob buys $\text{Padlock}_{\text{Bob}}$ and Key_{Bob}
- Bob sends $\text{Padlock}_{\text{Bob}}$ to Alice
- Alice puts the secret message in a box
- Alice locks the box using $\text{Padlock}_{\text{Bob}}$
- Alice sends the box to Bob
- Bob receives the box
- Bob unlocks the box using Key_{Bob}

Public-Key Cryptography

- S_{bank} is never *communicated*
- Single point of failure

Public-Key Cryptography

- $S_{\mathtt{bank}}$ is never *communicated*
- Single point of failure

Public-Key Cryptography

- $S_{\mathtt{bank}}$ is never *communicated*
- Single point of failure

The Most Criticial Aspect of any Security Measure †

- Not how well it works
- But how well it fails
  - INTEGRITY: Secret key can be lost
  - SECRECY: Secret key can be compromised

†C. C. Mann, “Homeland insecurity,” *The Atlantic Monthly*, vol. 290, pp. 81–102, September 2002 [2]

The Most Criticial Aspect of any Security Measure †

- Not how well it works
- But how well it fails
  - INTEGRITY: Secret key can be lost
  - SECRECY: Secret key can be compromised

†C. C. Mann, "Homeland insecurity," *The Atlantic Monthly*, vol. 290, pp. 81–102, September 2002 [2]

The Most Criticial Aspect of any Security Measure †

- Not how well it works
- But how well it fails
  - INTEGRITY: Secret key can be lost
  - SECRECY: Secret key can be compromised

† C. C. Mann, "Homeland insecurity," *The Atlantic Monthly*, vol. 290, pp. 81–102, September 2002 [2]

The Most Criticial Aspect of any Security Measure †

- Not how well it works
- But how well it fails
  - INTEGRITY: Secret key can be lost
  - SECRECY: Secret key can be compromised

†C. C. Mann, "Homeland insecurity," *The Atlantic Monthly*, vol. 290, pp. 81–102, September 2002 [2]

The Most Criticial Aspect of any Security Measure †

- Not how well it works
- But how well it fails
  - INTEGRITY: Secret key can be lost
  - SECRECY: Secret key can be compromised

†C. C. Mann, “Homeland insecurity,” *The Atlantic Monthly*, vol. 290, pp. 81–102, September 2002 [2]

Divergent Goals ‡

- Data Integrity
  - SOLUTION: Duplication of data among n parties would prevent coalitions of up to $n-1$ parties from erasing the secret
  - ISSUE: Any of the n parties could disclose the secret to an adversary
- Data Secrecy
  - SOLUTION: Splitting the data into n pieces would prevent full-disclosure from any single party
  - ISSUE: Destruction of any one piece could erase the secret

‡P. S. Gemmell, "An introduction to threshold cryptography," *CryptoBytes*, vol. 2, pp. 7–12, Winter 1997 [3]

Divergent Goals ‡

- Data Integrity
  - Solution: Duplication of data among n parties would prevent coalitions of up to $n-1$ parties from erasing the secret
  - Issue: Any of the n parties could disclose the secret to an adversary
- Data Secrecy
  - Solution: Splitting the data into n pieces would prevent full-disclosure from any single party
  - Issue: Destruction of any one piece could erase the secret

‡ P. S. Gemmell, "An introduction to threshold cryptography," *CryptoBytes*, vol. 2, pp. 7–12, Winter 1997 [3]

Divergent Goals ‡

- Data Integrity
  - SOLUTION: Duplication of data among n parties would prevent coalitions of up to $n-1$ parties from erasing the secret
  - ISSUE: Any of the n parties could disclose the secret to an adversary
- Data Secrecy
  - SOLUTION: Splitting the data into n pieces would prevent full-disclosure from any single party
  - ISSUE: Destruction of any one piece could erase the secret

‡P. S. Gemmell, "An introduction to threshold cryptography," *CryptoBytes*, vol. 2, pp. 7–12, Winter 1997 [3]

Divergent Goals ‡

- Data Integrity
  - SOLUTION: Duplication of data among n parties would prevent coalitions of up to $n-1$ parties from erasing the secret
  - ISSUE: Any of the n parties could disclose the secret to an adversary
- Data Secrecy
  - SOLUTION: Splitting the data into n pieces would prevent full-disclosure from any single party
  - ISSUE: Destruction of any one piece could erase the secret

‡P. S. Gemmell, “An introduction to threshold cryptography,” *CryptoBytes*, vol. 2, pp. 7–12, Winter 1997 [3]

Divergent Goals ‡

- Data Integrity
  - Solution: Duplication of data among n parties would prevent coalitions of up to $n-1$ parties from erasing the secret
  - Issue: Any of the n parties could disclose the secret to an adversary
- Data Secrecy
  - Solution: Splitting the data into n pieces would prevent full-disclosure from any single party
  - Issue: Destruction of any one piece could erase the secret

‡P. S. Gemmell, "An introduction to threshold cryptography," *CryptoBytes*, vol. 2, pp. 7–12, Winter 1997 [3]

Divergent Goals ‡

- Data Integrity
  - SOLUTION: Duplication of data among n parties would prevent coalitions of up to $n-1$ parties from erasing the secret
  - ISSUE: Any of the n parties could disclose the secret to an adversary
- Data Secrecy
  - SOLUTION: Splitting the data into n pieces would prevent full-disclosure from any single party
  - ISSUE: Destruction of any one piece could erase the secret

‡P. S. Gemmell, "An introduction to threshold cryptography," *CryptoBytes*, vol. 2, pp. 7–12, Winter 1997 [3]

Divergent Goals ‡

- Data Integrity
  - SOLUTION: Duplication of data among n parties would prevent coalitions of up to $n - 1$ parties from erasing the secret
  - ISSUE: Any of the n parties could disclose the secret to an adversary
- Data Secrecy
  - SOLUTION: Splitting the data into n pieces would prevent full-disclosure from any single party
  - ISSUE: Destruction of any one piece could erase the secret

‡P. S. Gemmell, "An introduction to threshold cryptography," *CryptoBytes*, vol. 2, pp. 7–12, Winter 1997 [3]

Implementation §

- Divide D into n pieces $D_1, \ldots, D_n$ in such a way that:
  - Knowledge of any k or more D_i pieces makes D easily computable
  - Knowledge of any $k - 1$ or fewer D_i pieces leaves D completely undetermined
- Example: $(3, n)$ threshold scheme for signatures on a check
  - An unfaithful executive must have at least two accomplices in order to forge a valid signature

§A. Shamir, "How to share a secret," *Communications of the Association for Computing Machinery*, vol. 22, pp. 612–613, Nov. 1979 [4]

Implementation §

- Divide D into n pieces $D_1, ..., D_n$ in such a way that:
  - Knowledge of any k or more D_i pieces makes D easily computable
  - Knowledge of any $k-1$ or fewer D_i pieces leaves D completely undetermined
- Example: $(3, n)$ threshold scheme for signatures on a check
  - An unfaithful executive must have at least two accomplices in order to forge a valid signature

§ A. Shamir, "How to share a secret," *Communications of the Association for Computing Machinery*, vol. 22, pp. 612–613, Nov. 1979 [4]

Implementation §

- Divide D into n pieces $D_1, \ldots, D_n$ in such a way that:
  - Knowledge of any k or more D_i pieces makes D easily computable
  - Knowledge of any $k - 1$ or fewer D_i pieces leaves D completely undetermined
- Example: $(3, n)$ threshold scheme for signatures on a check
  - An unfaithful executive must have at least two accomplices in order to forge a valid signature

§ A. Shamir, "How to share a secret," *Communications of the Association for Computing Machinery*, vol. 22, pp. 612–613, Nov. 1979 [4]

Implementation §

- Divide D into n pieces $D_1, ..., D_n$ in such a way that:
  - Knowledge of any k or more D_i pieces makes D easily computable
  - Knowledge of any $k - 1$ or fewer D_i pieces leaves D completely undetermined
- Example: $(3, n)$ threshold scheme for signatures on a check
  - An unfaithful executive must have at least two accomplices in order to forge a valid signature

§ A. Shamir, "How to share a secret," *Communications of the Association for Computing Machinery*, vol. 22, pp. 612–613, Nov. 1979 [4]

Implementation §

- Divide D into n pieces $D_1, ..., D_n$ in such a way that:
  - Knowledge of any k or more D_i pieces makes D easily computable
  - Knowledge of any $k - 1$ or fewer D_i pieces leaves D completely undetermined
- Example: $(3, n)$ threshold scheme for signatures on a check
  - An unfaithful executive must have at least two accomplices in order to forge a valid signature

§A. Shamir, "How to share a secret," *Communications of the Association for Computing Machinery*, vol. 22, pp. 612–613, Nov. 1979 [4]

Implementation §

- Divide D into n pieces $D_1, ..., D_n$ in such a way that:
  - Knowledge of any k or more D_i pieces makes D easily computable
  - Knowledge of any $k - 1$ or fewer D_i pieces leaves D completely undetermined
- Example: $(3, n)$ threshold scheme for signatures on a check
  - An unfaithful executive must have at least two accomplices in order to forge a valid signature

§A. Shamir, "How to share a secret," *Communications of the Association for Computing Machinery*, vol. 22, pp. 612–613, Nov. 1979 [4]

Previous Solutions

- Data Integrity
  - SOLUTION: Duplication of data among n parties would prevent coalitions of up to $n-1$ parties from erasing the secret
  - THRESHOLD VALUES: $(1, n)$
- Data Secrecy
  - SOLUTION: Splitting the data into n pieces would prevent full-disclosure from any single party
  - THRESHOLD VALUES: (n, n)

Previous Solutions

- Data Integrity
  - SOLUTION: Duplication of data among n parties would prevent coalitions of up to $n - 1$ parties from erasing the secret
  - THRESHOLD VALUES: $(1, n)$
- Data Secrecy
  - SOLUTION: Splitting the data into n pieces would prevent full-disclosure from any single party
  - THRESHOLD VALUES: (n, n)

Previous Solutions

- Data Integrity
  - SOLUTION: Duplication of data among n parties would prevent coalitions of up to $n - 1$ parties from erasing the secret
  - THRESHOLD VALUES: $(1, n)$
- Data Secrecy
  - SOLUTION: Splitting the data into n pieces would prevent full-disclosure from any single party
  - THRESHOLD VALUES: (n, n)

Previous Solutions

- Data Integrity
  - SOLUTION: Duplication of data among n parties would prevent coalitions of up to $n - 1$ parties from erasing the secret
  - THRESHOLD VALUES: $(1, n)$
- Data Secrecy
  - SOLUTION: Splitting the data into n pieces would prevent full-disclosure from any single party
  - THRESHOLD VALUES: (n, n)

Previous Solutions

- Data Integrity
  - Solution: Duplication of data among n parties would prevent coalitions of up to $n - 1$ parties from erasing the secret
  - Threshold Values: $(1, n)$
- Data Secrecy
  - Solution: Splitting the data into n pieces would prevent full-disclosure from any single party
  - Threshold Values: (n, n)

Previous Solutions

- Data Integrity
  - SOLUTION: Duplication of data among n parties would prevent coalitions of up to $n - 1$ parties from erasing the secret
  - THRESHOLD VALUES: $(1, n)$
- Data Secrecy
  - SOLUTION: Splitting the data into n pieces would prevent full-disclosure from any single party
  - THRESHOLD VALUES: (n, n)

Previous Solutions

- Data Integrity
  - SOLUTION: Duplication of data among n parties would prevent coalitions of up to $n-1$ parties from erasing the secret
  - THRESHOLD VALUES: $(1, n)$
- Data Secrecy
  - SOLUTION: Splitting the data into n pieces would prevent full-disclosure from any single party
  - THRESHOLD VALUES: (n, n)

Implementation

By properly choosing k and n parameters we can give:

- Any sufficiently large majority (k) the authority to do some action
- Any sufficiently large minority ($n - k + 1$) the power to block it

Implementation

By properly choosing k and n parameters we can give:

- Any sufficiently large majority (k) the authority to do some action
- Any sufficiently large minority ($n - k + 1$) the power to block it

Implementation

By properly choosing k and n parameters we can give:

- Any sufficiently large majority (k) the authority to do some action
- Any sufficiently large minority ($n - k + 1$) the power to block it

Advantages

By using a (k, n) threshold scheme with $n = 2k - 1$:

- We can recover the original key even when $[\frac{n}{2}] = k - 1$ of the n pieces are destroyed.
- Opponents cannot reconstruct the key even when a security breach exposes $[\frac{n}{2}] = k - 1$ of the remaining k pieces.

Advantages

By using a (k, n) threshold scheme with $n = 2k - 1$:

- We can recover the original key even when $[\frac{n}{2}] = k - 1$ of the n pieces are destroyed.
- Opponents cannot reconstruct the key even when a security breach exposes $[\frac{n}{2}] = k - 1$ of the remaining k pieces.

Advantages

By using a (k, n) threshold scheme with $n = 2k - 1$:

- We can recover the original key even when $[\frac{n}{2}] = k - 1$ of the n pieces are destroyed.
- Opponents cannot reconstruct the key even when a security breach exposes $[\frac{n}{2}] = k - 1$ of the remaining k pieces.

Disadvantages

Inconvenience:

- $(1, 9)$ is convenient but easy to misuse
- $(5, 9)$ is safe but inconvenient

Disadvantages

Inconvenience:

- $(1, 9)$ is convenient but easy to misuse
- $(5, 9)$ is safe but inconvenient

Disadvantages

Inconvenience:

- $(1, 9)$ is convenient but easy to misuse
- $(5, 9)$ is safe but inconvenient

Threshold Cryptography Software

Create software for implementing a (k, n) threshold scheme in cryptographic aspects of a Certificate Authority and Web Server

Threshold Cryptography Software

Create software for implementing a (k, n) *threshold scheme in cryptographic aspects of a Certificate Authority and Web Server*

Single-Secret Sharing

- LaGrange Interpolation [4]
- Intersecting Hyperplanes ¶
- Combinations of Families and Committees ‖

¶G. R. Blakley, "Safeguarding cryptographic keys," in *1979 National Computer Conference: June 4–7, 1979, New York, New York* (R. E. Merwin, J. T. Zanca, and M. Smith, eds.), vol. 48 of *AFIPS Conference proceedings*, (Montvale, NJ, USA), pp. 313–317, AFIPS Press, 1979 [5]

‖N. Alon, Z. Galil, and M. Yung, "Dynamic re-sharing verifiable secret sharing against a mobile adversary," in *Algorithms — ESA '95: Third Annual European Symposium, Corfu, Greece, September 25–27, 1995: proceedings* (P. G. Spirakis, ed.), vol. 979 of *Lecture Notes in Computer Science*, (Berlin, Germany / Heidelberg, Germany / London, UK / etc.), pp. 523–537, Springer-Verlag, 1995 [6]

Single-Secret Sharing

- LaGrange Interpolation [4]
- Intersecting Hyperplanes ¶
- Combinations of Families and Committees ‖

¶G. R. Blakley, "Safeguarding cryptographic keys," in *1979 National Computer Conference: June 4–7, 1979, New York, New York* (R. E. Merwin, J. T. Zanca, and M. Smith, eds.), vol. 48 of *AFIPS Conference proceedings*, (Montvale, NJ, USA), pp. 313–317, AFIPS Press, 1979 [5]

‖N. Alon, Z. Galil, and M. Yung, "Dynamic re-sharing verifiable secret sharing against a mobile adversary," in *Algorithms — ESA '95: Third Annual European Symposium, Corfu, Greece, September 25–27, 1995: proceedings* (P. G. Spirakis, ed.), vol. 979 of *Lecture Notes in Computer Science*, (Berlin, Germany / Heidelberg, Germany / London, UK / etc.), pp. 523–537, Springer-Verlag, 1995 [6]

Single-Secret Sharing

- LaGrange Interpolation [4]
- Intersecting Hyperplanes [¶]
- Combinations of Families and Committees [‖]

[¶]G. R. Blakley, "Safeguarding cryptographic keys," in *1979 National Computer Conference: June 4–7, 1979, New York, New York* (R. E. Merwin, J. T. Zanca, and M. Smith, eds.), vol. 48 of *AFIPS Conference proceedings*, (Montvale, NJ, USA), pp. 313–317, AFIPS Press, 1979 [5]

[‖]N. Alon, Z. Galil, and M. Yung, "Dynamic re-sharing verifiable secret sharing against a mobile adversary," in *Algorithms — ESA '95: Third Annual European Symposium, Corfu, Greece, September 25–27, 1995: proceedings* (P. G. Spirakis, ed.), vol. 979 of *Lecture Notes in Computer Science*, (Berlin, Germany / Heidelberg, Germany / London, UK / etc.), pp. 523–537, Springer-Verlag, 1995 [6]

Single-Secret Sharing

- LaGrange Interpolation [4]
- Intersecting Hyperplanes ¶
- Combinations of Families and Committees ‖

¶G. R. Blakley, "Safeguarding cryptographic keys," in *1979 National Computer Conference: June 4–7, 1979, New York, New York* (R. E. Merwin, J. T. Zanca, and M. Smith, eds.), vol. 48 of *AFIPS Conference proceedings*, (Montvale, NJ, USA), pp. 313–317, AFIPS Press, 1979 [5]

‖N. Alon, Z. Galil, and M. Yung, "Dynamic re-sharing verifiable secret sharing against a mobile adversary," in *Algorithms — ESA '95: Third Annual European Symposium, Corfu, Greece, September 25–27, 1995: proceedings* (P. G. Spirakis, ed.), vol. 979 of *Lecture Notes in Computer Science*, (Berlin, Germany / Heidelberg, Germany / London, UK / etc.), pp. 523–537, Springer-Verlag, 1995 [6]

Cryptographic Function Sharing

- Any k shareholders should be able to collectively compute f.
- Even after taking part in the computation of f on some inputs, no set of upto $k-1$ shareholders should be able to compute f on other inputs [3].

Cryptographic Function Sharing

- Any k shareholders should be able to collectively compute f.
- Even after taking part in the computation of f on some inputs, no set of upto $k-1$ shareholders should be able to compute f on other inputs [3].

Cryptographic Function Sharing

- Any k shareholders should be able to collectively compute f.
- Even after taking part in the computation of f on some inputs, no set of upto $k - 1$ shareholders should be able to compute f on other inputs [3].

RSA Sharing Protocols

- A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung, "How to share a function securely," in *Proceedings of the twenty-sixth annual ACM Symposium on the Theory of Computing: Montréal, Québec, Canada, May 23–25, 1994* (ACM, ed.), (New York, NY 10036, USA), pp. 522–533, ACM Press, 1994. ACM order no. 508930 [7]

RSA Sharing Protocols

- R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust and efficient sharing of RSA functions," in *Advances in cryptology, CRYPTO '96: 16th annual international cryptology conference, Santa Barbara, California, USA, August 18–22, 1996: proceedings* (N. Koblitz, ed.), vol. 1109 of *Lecture Notes in Computer Science*, (Berlin, Germany / Heidelberg, Germany / London, UK / etc.), pp. 157–172, Springer-Verlag, 1996. Sponsored by the International Association for Cryptologic Research (IACR), in cooperation with the IEEE Computer Society Technical Committee on Security and Privacy and the Computer Science Department of the University of California at Santa Barbara (UCSB) [8]

Deliverables

- ITTC Daemon
- Interface Library (`libittc.so`)
- OpenSSL Modifications
- lighttpd Modifications
- A production-ready combination of function sharing threshold cryptographic Certificate Authority and Web Server

Deliverables

- ITTC Daemon
- Interface Library (`libittc.so`)
- OpenSSL Modifications
- lighttpd Modifications
- A production-ready combination of function sharing threshold cryptographic Certificate Authority and Web Server

Deliverables

- ITTC Daemon
- Interface Library (`libittc.so`)
- OpenSSL Modifications
- lighttpd Modifications
- A production-ready combination of function sharing threshold cryptographic Certificate Authority and Web Server

Deliverables

- ITTC Daemon
- Interface Library (`libittc.so`)
- OpenSSL Modifications
- lighttpd Modifications
- A production-ready combination of function sharing threshold cryptographic Certificate Authority and Web Server

Deliverables

- ITTC Daemon
- Interface Library (`libittc.so`)
- OpenSSL Modifications
- lighttpd Modifications
- A production-ready combination of function sharing threshold cryptographic Certificate Authority and Web Server

Deliverables

- ITTC Daemon
- Interface Library (`libittc.so`)
- OpenSSL Modifications
- lighttpd Modifications
- A production-ready combination of function sharing threshold cryptographic Certificate Authority and Web Server

Wu, Malkin and Boneh's Implementation **

- SSLeay Modifications
- Apache Modifications

**T. Wu, M. Malkin, and D. Boneh, "Building intrusion tolerant applications," in *Proceedings of the 8th conference on USENIX Security Symposium*, (Berkeley, CA, USA), pp. 7–7, USENIX Association, 1999 [9]

Wu, Malkin and Boneh's Implementation **

- SSLeay Modifications
- Apache Modifications

**T. Wu, M. Malkin, and D. Boneh, "Building intrusion tolerant applications," in *Proceedings of the 8th conference on USENIX Security Symposium*, (Berkeley, CA, USA), pp. 7–7, USENIX Association, 1999 [9]

Wu, Malkin and Boneh's Implementation **

- SSLeay Modifications
- Apache Modifications

**T. Wu, M. Malkin, and D. Boneh, "Building intrusion tolerant applications," in *Proceedings of the 8th conference on USENIX Security Symposium*, (Berkeley, CA, USA), pp. 7–7, USENIX Association, 1999 [9]

Bibliography I

[1] N. Ferguson and B. Schneier, *Practical Cryptography*. New York, NY, USA: John Wiley & Sons, Inc., 2003.

[2] C. C. Mann, "Homeland insecurity," *The Atlantic Monthly*, vol. 290, pp. 81–102, September 2002.

[3] P. S. Gemmell, "An introduction to threshold cryptography," *CryptoBytes*, vol. 2, pp. 7–12, Winter 1997.

[4] A. Shamir, "How to share a secret," *Communications of the Association for Computing Machinery*, vol. 22, pp. 612–613, Nov. 1979.

Bibliography II

[5] G. R. Blakley, "Safeguarding cryptographic keys," in *1979 National Computer Conference: June 4–7, 1979, New York, New York* (R. E. Merwin, J. T. Zanca, and M. Smith, eds.), vol. 48 of *AFIPS Conference proceedings*, (Montvale, NJ, USA), pp. 313–317, AFIPS Press, 1979.

[6] N. Alon, Z. Galil, and M. Yung, "Dynamic re-sharing verifiable secret sharing against a mobile adversary," in *Algorithms — ESA '95: Third Annual European Symposium, Corfu, Greece, September 25–27, 1995: proceedings* (P. G. Spirakis, ed.), vol. 979 of *Lecture Notes in Computer Science*, (Berlin, Germany / Heidelberg, Germany / London, UK / etc.), pp. 523–537, Springer-Verlag, 1995.

Bibliography III

[7] A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung, "How to share a function securely," in *Proceedings of the twenty-sixth annual ACM Symposium on the Theory of Computing: Montréal, Québec, Canada, May 23–25, 1994* (ACM, ed.), (New York, NY 10036, USA), pp. 522–533, ACM Press, 1994. ACM order no. 508930.

[8] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust and efficient sharing of RSA functions," in *Advances in cryptology, CRYPTO '96: 16th annual international cryptology conference, Santa Barbara, California, USA, August 18–22, 1996: proceedings* (N. Koblitz, ed.), vol. 1109 of *Lecture Notes in Computer Science*, (Berlin, Germany / Heidelberg,

Bibliography IV

Germany / London, UK / etc.), pp. 157–172, Springer-Verlag, 1996. Sponsored by the International Association for Cryptologic Research (IACR), in cooperation with the IEEE Computer Society Technical Committee on Security and Privacy and the Computer Science Department of the University of California at Santa Barbara (UCSB).

[9] T. Wu, M. Malkin, and D. Boneh, "Building intrusion tolerant applications," in *Proceedings of the 8th conference on USENIX Security Symposium*, (Berkeley, CA, USA), pp. 7–7, USENIX Association, 1999.

Questions

Questions are never indiscreet: answers sometimes are.
(Oscar Wilde)