Unconditionally Secure Cryptography

Douglas R. Stinson

David R. Cheriton School of Computer Science
University of Waterloo

CS 858
Spring 2019

Outline

1. Introduction to unconditional security (aka information theoretic security)
2. Encryption schemes and perfect secrecy (**Shannon's theory**)
3. Message authentication (**Wegman-Carter universal hashing**)
4. Secret sharing (**Shamir's Scheme**)
5. Key predistribution (**Blom's scheme**)

What does the term "secure" mean?

- In the August 1977 issue of **Scientific American**, Martin Gardner wrote a column on the newly developed RSA public-key cryptosystem entitled **"A new kind of cipher that would take millions of years to break"**. Included in the article was a challenge ciphertext, encrypted using a 512-bit RSA key.
- The challenge was solved 17 years later, on April 26, 1994, by Arjen Lenstra *et al*.
- They factored the given public key and decrypted the ciphertext to yield the plaintext, which was **"The Magic Words are Squeamish Ossifrage"**.
- For this, they claimed a prize of U.S. $100.

What Went Wrong

The statement that the cipher would take millions of years to break probably referred to how long it would take to run the best factoring algorithm known in 1977 on the fastest computer available in 1977. However, between 1977 and 1994:

- **computers became much faster**
- **improved factoring algorithms were found**
- **the development of the internet facilitated large-scale distributed computations**.

Even so, the factorization still required over 5000 MIPS-years of computation time in 1994.
The current state-of-the-art is the factorization of the 232-digit (768 bit) challenge, RSA-768, in December 2009.

A Formal Model for Security

Any discussion of cryptographic security requires a specification of an attack model, computational resources, and an adversarial goal. These terms are defined as follows:

attack model

The attack model specifies the **information available to the adversary**. In the case of encryption schemes, this information could include ciphertexts, plaintext-ciphertext pairs, temporary access to encryption or decryption oracles, etc. We will always assume that the adversary knows the protocol being used (this is called **Kerckhoff's Principle**).

A Formal Model (cont.)

computational resources

Here, we specify the computational resources available to the adversary, such as computing equipment, algorithms, computing time, etc. In the case of **unconditional security** (aka **information theoretic security**), the adversary is assumed to have unlimited computational resources.

adversarial goal

The adversarial goal specifies what it means to "break" the protocol. What is the adversary attempting to do and/or what problem is he trying to solve? How is the notion of a "successful attack" defined?

A Formal Model (cont.)

A statement of security for a cryptographic scheme will be an assertion of the following form:

In attack model $\mathcal{A}$ and given specified computational resources $\mathcal{C}$, a particular adversarial goal $\mathcal{G}$ cannot be achieved with probability exceeding some specified value ϵ.

As mentioned before, in the setting of unconditional security, there is no limit on the computational resources available to the adversary.

What is (Im)Possible in an Unconditional Secure Setting?

- Secure **secret-key encryption schemes** exist, but secure public-key encryption schemes do not exist.
- Secure **message authentication codes** exist.
- Secure **signature schemes** exist, but not in the public key setting — secret verification keys are required, and these kinds of schemes only work for a fixed number of participants.
- Secure **key distribution schemes** exist, but secure key agreement is impossible without prior shared secrets.
- Secure **commitment schemes** exist if there is a trusted initializer.
- Secure **secret sharing schemes** exist.

Issues and Goals in Unconditional Security

- Typically, unconditionally secure schemes can be used only for a **fixed number of times** (e.g., the **One-time Pad** can be used for only one encryption).
- Unconditionally secure schemes often require a **trusted authority** to set up the scheme.
- The main goal in unconditional security is to balance security against memory requirements (storage) and/or communication complexity (amount of information transmitted).
- In a multiuser setting, security guarantees only hold against **coalitions of a prespecified fixed size**.

Cryptosystems (encryption schemes)

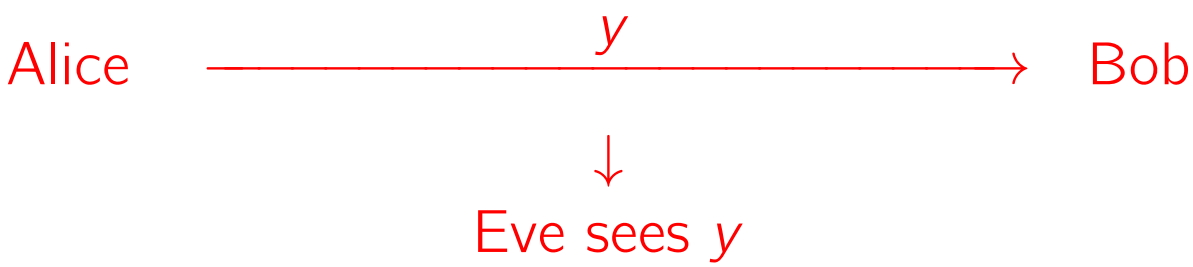
Definition

A **cryptosystem** is a five-tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where the following conditions are satisfied:

1. $\mathcal{P}$ is a finite set of possible **plaintexts**.
2. $\mathcal{C}$ is a finite set of possible **ciphertexts**.
3. $\mathcal{K}$, the **keyspace**, is a finite set of possible **keys**.
4. For each $K \in \mathcal{K}$, there is an **encryption rule** $e_K \in \mathcal{E}$ and a corresponding **decryption rule** $d_K \in \mathcal{D}$. Each $e_K : \mathcal{P} \to \mathcal{C}$ and $d_K : \mathcal{C} \to \mathcal{P}$ are functions such that $d_K(e_K(x)) = x$ for every plaintext element $x \in \mathcal{P}$.

Communication Channel

- Alice and Bob have a shared secret key K and Alice computes $y = e_K(x)$ to encrypt the plaintext x.
- Bob computes $x = d_K(y)$ to decrypt the ciphertext.
- The ciphertext y is observed by the **passive adversary** Eve.
- However, Eve does not know K.
- Hopefully, Eve cannot figure out what K or x are.



Perfect Secrecy

- Assume that a cryptosystem $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is specified, and a particular key $K \in \mathcal{K}$ is used for **only one encryption**.
- There is a probability distribution on the plaintext space, $\mathcal{P}$.
- Denote the **a priori** probability that plaintext x occurs by $\Pr[\mathbf{x} = x]$.
- The key K is chosen (using some fixed probability distribution).
- Denote the probability that key K is chosen by $\Pr[\mathbf{K} = K]$.
- The key and the plaintext are independent random variables.
- The two probability distributions on $\mathcal{P}$ and $\mathcal{K}$ induce a probability distribution on $\mathcal{C}$.

Perfect Secrecy (cont.)

- We compute the probability $\Pr[\mathbf{y} = y]$ that y is the ciphertext that is transmitted.
- For a key $K \in \mathcal{K}$, define

$$C(K) = \{e_K(x) : x \in \mathcal{P}\}.$$

  $C(K)$ represents the set of **possible ciphertexts** if K is the key.
- For every $y \in \mathcal{C}$, we have that

$$\Pr[\mathbf{y} = y] = \sum_{\{K : y \in C(K)\}} \Pr[\mathbf{K} = K] \Pr[\mathbf{x} = d_K(y)].$$

- For any $y \in \mathcal{C}$ and $x \in \mathcal{P}$, we can compute

$$\Pr[\mathbf{y} = y | \mathbf{x} = x] = \sum_{\{K : x = d_K(y)\}} \Pr[\mathbf{K} = K].$$

Perfect Secrecy (cont.)

- $\Pr[\mathbf{x} = x | \mathbf{y} = y]$ can be computed using **Bayes' theorem**:

$$\Pr[\mathbf{x} = x | \mathbf{y} = y] = \frac{\Pr[\mathbf{x} = x] \times \Pr[\mathbf{y} = y | \mathbf{x} = x]}{\Pr[\mathbf{y} = y]}.$$

- **Perfect secrecy** means that $\Pr[x|y] = \Pr[x]$ for all x, y.
- Perfectly secrecy provides unconditional security in a **known-ciphertext attack** if K is used for only one encryption.
- The attack model is that the adversary Eve is given a ciphertext, y.
- Eve's goal is to compute information about the plaintext of x.

An Example

Let $\mathcal{P} = \{a, b\}$ with

$$\Pr[a] = 1/4, \Pr[b] = 3/4.$$

Let $\mathcal{K} = \{K_1, K_2, K_3\}$ with

$$\Pr[K_1] = 1/2, \Pr[K_2] = \Pr[K_3] = 1/4.$$

Let $\mathcal{C} = \{1, 2, 3, 4\}$, and suppose the encryption functions are defined by the following **encryption matrix**:

	a	b
K_1	1	2
K_2	2	3
K_3	3	4

An Example (cont.)

We first compute the probability distribution on $\mathcal{C}$. We obtain the following:

$$\begin{aligned}
\Pr[1] &= \frac{1}{8} \\
\Pr[2] &= \frac{3}{8} + \frac{1}{16} = \frac{7}{16} \\
\Pr[3] &= \frac{3}{16} + \frac{1}{16} = \frac{1}{4} \\
\Pr[4] &= \frac{3}{16}.
\end{aligned}$$

An Example (cont.)

Next, we can compute the conditional probability distributions on the ciphertexts, given a plaintext. We have:

$$\begin{aligned}
\Pr[1|a] &= \frac{1}{2} & \Pr[1|b] &= 0 \\
\Pr[2|a] &= \frac{1}{4} & \Pr[2|b] &= \frac{1}{2} \\
\Pr[3|a] &= \frac{1}{4} & \Pr[3|b] &= \frac{1}{4} \\
\Pr[4|a] &= 0 & \Pr[4|b] &= \frac{1}{4}.
\end{aligned}$$

An Example (cont.)

Finally, we use Bayes' Theorem to compute the conditional probability distributions on the plaintext, given a ciphertext:

$$\begin{aligned}
\Pr[a|1] &= 1 & \Pr[b|1] &= 0 \\
\Pr[a|2] &= \frac{1}{7} & \Pr[b|2] &= \frac{6}{7} \\
\Pr[a|3] &= \frac{1}{4} & \Pr[b|3] &= \frac{3}{4} \\
\Pr[a|4] &= 0 & \Pr[b|4] &= 1.
\end{aligned}$$

Recalling that $\Pr[a] = 1/4$ and $\Pr[b] = 3/4$, we see that the perfect secrecy property is satisfied for the ciphertext $y = 3$, but not for the other three ciphertexts.

One-time Pad

The one-time pad is the best known example of a cryptosystem that provides perfect secrecy.

Protocol: One-time Pad

Let $n \geq 1$ be an integer, and take $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$. For every $K \in (\mathbb{Z}_2)^n$, let $\mathsf{Pr}[K] = 2^{-n}$ (i.e., keys are chosen equiprobably) and for every $x \in (\mathbb{Z}_2)^n$, define $e_K(x)$ to be the vector sum modulo 2 of K and x (or, equivalently, the exclusive-or of the two associated bitstrings). So, if $x = (x_1, \ldots, x_n)$ and $K = (K_1, \ldots, K_n)$, then

$$e_K(x) = x \oplus K = (x_1 + K_1, \ldots, x_n + K_n) \bmod 2.$$

Decryption is identical to encryption. If $y = (y_1, \ldots, y_n)$, then

$$d_K(y) = y \oplus K = (y_1 + K_1, \ldots, y_n + K_n) \bmod 2.$$

An Example

Suppose $n = 2$. The encryption matrix for the one-time pad is as follows:

K	$x = 00$	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

Security of the One-time Pad

It is simple to prove that the **One-time Pad** provides perfect secrecy:

- For every $K \in (\mathbb{Z}_2)^n$,

$$\Pr[K] = 2^{-n}.$$

- For every $x, y \in (\mathbb{Z}_2)^n$,

$$\Pr[y|x] = \Pr[\mathbf{K} = x \oplus y] = 2^{-n}.$$

- For every $y \in (\mathbb{Z}_2)^n$,

$$\Pr[y] = \sum_{x \in (\mathbb{Z}_2)^n} (\Pr[x] \times \Pr[y|x]) = \sum_{x \in (\mathbb{Z}_2)^n} (\Pr[x] \times 2^{-n}) = 2^{-n}.$$

- Now use **Bayes' Theorem** to compute $\Pr[x|y]$:

$$\Pr[x|y] = \frac{\Pr[y|x] \times \Pr[x]}{\Pr[y]} = \frac{2^{-n} \times \Pr[x]}{2^{-n}} = \Pr[x].$$

Combinatorial Characterization

The following results are due to Shannon.

Theorem

Suppose $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *is a cryptosystem.*

1. *If* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *provides perfect secrecy, then* $|\mathcal{K}| \geq |\mathcal{C}| \geq |\mathcal{P}|$.
2. *Suppose that* $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{P}|$. *Then* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *provides perfect secrecy if and only if every key is used with equal probability* $1/|\mathcal{K}|$, *and for every* $x \in \mathcal{P}$ *and every* $y \in \mathcal{C}$, *there is a unique key* K *such that* $e_K(x) = y$. *(That is, the encryption matrix is a* **Latin square**.*)*

A Latin Square

Here is a Latin square of order 5:

1	2	3	4	5
2	3	4	5	1
3	4	5	1	2
4	5	1	2	3
5	1	2	3	4

Weakening the Perfect Secrecy Requirement

Some results have been proven about encryption schemes with "short" keys that weaken the perfect secrecy requirement, e.g., in Y. Dodis and A. Smith, **"Entropic security and the encryption of high entropy messages"**, *LNCS* **3378** (2005), 556–577.

Theorem

Suppose $\mathcal{P} = \{0,1\}^n$ *has a probability distribution such that* $\Pr[x] \leq 2^{-t}$ *for all* $x \in \mathcal{P}$. *Let* $k = n - t + 2\log_2(1/\epsilon) + 2$ *for some positive real number* ϵ, *where* $k \leq n$. *Define* $\mathcal{K} = \{0,1\}^k$. *For* $x \in \mathcal{P}$ *and* $K \in \mathcal{K}$, *define* $y = e_K(x) = (r, x \oplus rK)$, *where* $r \in \{0,1\}^n$ *is chosen randomly and* rK *is computed in* $\mathbb{F}_{2^n}$. *Let* f *be any function with domain* $\mathcal{P}$. *Then, given* y, *no adversary can predict* $f(x)$ *with advantage exceeding* ϵ.

"Advantage" refers to the **increase** in the probability that the adversary can compute $f(x)$ after he is given y.

Authentication Codes

Definition

An **authentication code** is a four-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \mathcal{H})$, where the following conditions are satisfied:

1. $\mathcal{S}$ is a finite set of possible **source states**.
2. $\mathcal{A}$ is a finite set of possible **authentication tags**.
3. $\mathcal{K}$, the **keyspace**, is a finite set of possible **keys**.
4. $\mathcal{H}$ is a finite set of possible **authentication rules**. For each $K \in \mathcal{K}$, there is an authentication rule $h_K \in \mathcal{H}$ such that $h_K : \mathcal{S} \to \mathcal{A}$.

The **message set** is defined to be $\mathcal{M} = \mathcal{S} \times \mathcal{A}$.
A message $m = (s, a)$ is **valid** under key K if $h_K(s) = a$.

Simmons' Model for Authentication

In an **impersonation attack**, the **active adversary**, Oscar, introduces a message (s, a) into the channel, hoping it is valid:

$$\text{Oscar} \xrightarrow{(s,a)} \text{Bob}$$

In a **substitution attack**, Oscar observes a message (s, a), and then replaces it with a new message (s', a'), where $s' \neq s$, hoping (s', a') is valid:

$$\text{Alice} \xrightarrow{(s,a)} \text{Oscar} \xrightarrow{(s',a')} \text{Bob}$$

Deception Probabilities

- Assume there are known probability distributions on $\mathcal{S}$ and $\mathcal{K}$.
- Oscar's optimal strategy for impersonation yields a **impersonation probability** denoted by P_{d_0}.
- Oscar optimal strategy for substitution yields a **substitution probability** denoted by P_{d_1}.
- It is easy to see that $P_{d_0} \geq 1/|\mathcal{A}|$ and $P_{d_1} \geq 1/|\mathcal{A}|$.
- Oscar can always do at least this well by making random guess. (Optimal strategies may be better, however.)

An Example

	0	1	2
K_1	0	0	0
K_2	1	1	1
K_3	2	2	2
K_4	0	1	2
K_5	1	2	0
K_6	2	0	1
K_7	0	2	1
K_8	1	0	2
K_9	2	1	0

Suppose $\mathcal{S} = \mathcal{A} = \mathbb{Z}_3$ and $\mathcal{K} = \{K_1, \ldots, K_9\}$. We list all the authentication rules in an **authentication matrix**, and we suppose that every key is used with probability $1/9$.

An Example (cont.)

- Every tag is valid under three of the nine authentication rules.
- This immediately implies that $P_{d_0} = 3/9 = 1/3$.
- Given any valid message (s, a), the number of possible keys is reduced from nine to three.
- However, any other message (s', a') (where $s' \neq s$), will be valid under exactly one of these three keys.
- Therefore, $P_{d_1} = 1/3$.

Almost Strongly Universal Hashing

Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. A function $h : \mathcal{X} \to \mathcal{Y}$ will be termed a **hash function**. Let $\mathcal{H}$ be a set of hash functions from $\mathcal{X}$ to $\mathcal{Y}$. Let ϵ be a positive real number. $\mathcal{H}$ is **ϵ-almost strongly-universal** (or **ϵ-ASU**) if the following two conditions are satisfied:

1. For every $x_1 \in \mathcal{X}$ and for every $y_1 \in \mathcal{Y}$,

$$|\{h \in \mathcal{H} : h(x_1) = y_1\}| = \frac{|\mathcal{H}|}{|\mathcal{Y}|}.$$

2. For every $x_1, x_2 \in \mathcal{X}$ $(x_1 \neq x_2)$ and for every $y_1, y_2 \in \mathcal{Y}$,

$$|\{h \in \mathcal{H} : h(x_1) = y_1, h(x_2) = y_2\}| \leq \frac{\epsilon|\mathcal{H}|}{|\mathcal{Y}|}.$$

We denote $\mathcal{H}$ as an $(N; n, m)$-**ϵ-ASU hash family**, where $|\mathcal{H}| = N$, $|\mathcal{X}| = n$ and $|\mathcal{Y}| = m$.

Almost Strongly Universal Hashing (cont.)

Properties 1. and 2. can be rephrased as follows:

1. For every $x_1 \in \mathcal{X}$ and for every $y_1 \in \mathcal{Y}$,

$$\Pr[h(x_1) = y_1] = \frac{1}{|\mathcal{Y}|}.$$

2. For every $x_1, x_2 \in \mathcal{X}$ $(x_1 \neq x_2)$ and for every $y_1, y_2 \in \mathcal{Y}$,

$$\Pr[h(x_1) = y_1 \wedge h(x_2) = y_2] \leq \frac{\epsilon}{|\mathcal{Y}|}.$$

Observe that 1. and 2. imply the following:

3. For every $x_1, x_2 \in \mathcal{X}$ $(x_1 \neq x_2)$ and for every $y_1, y_2 \in \mathcal{Y}$,

$$\Pr[h(x_2) = y_2 | h(x_1) = y_1] \leq \epsilon.$$

All probabilities are computed over a random choice of $h \in \mathcal{H}$.

Universal Hashing and Authentication Codes

Theorem

If there exists an $(N; n, m)$*-*ϵ*-ASU family* $|\mathcal{H}|$ *of hash functions from* $\mathcal{X}$ *to* $\mathcal{Y}$*, then there exists an authentication code for* n *source states, having* m *authentication tags and* N *authentication rules (and keys), such that* $P_{d_0} = 1/m$ *and* $P_{d_1} \leq \epsilon$*.*

Source states correspond to elements of $\mathcal{X}$, authentication tags correspond to elements of $\mathcal{Y}$ and authentication rules correspond to the hash functions in $\mathcal{H}$.

Strongly Universal Hashing

- In any $(N; n, m)$-ϵ-ASU hash family, $\epsilon \geq 1/m$.
- The hash family is **strongly-universal** (or **SU**) if $\epsilon = 1/m$.
- The previous example was a $(9; 3, 3)$-SU hash family.
- Strongly universal hash families are equivalent to combinatorial structures known as **orthogonal arrays**.
- An $(N; n, m)$-SU hash family is equivalent to an $\mathrm{OA}_\lambda(n, m)$, where $\lambda = N/m^2$.
- A classical bound for orthogonal arrays states that $\lambda \geq (n(m-1)+1)/m^2$.
- In the corresponding authentication code, $N \geq n(m-1)+1 \approx nm$, so $\log_2 |\mathcal{K}| \gtrapprox \log_2 |\mathcal{S}| + \log_2 |\mathcal{A}|$.

Wegman-Carter Universal Hashing

- The previous result says that the key is **very long** if we use a strongly universal hash family for authentication.
- In fact, an identical bound holds for any authentication code that attains the optimal (i.e., minimum) values of P_{d_0} and P_{d_1}.
- Wegman and Carter gave a construction that showed that the key length could be **reduced dramatically** if P_{d_1} is a **bit bigger** than the optimal value.
- Here we describe a general framework to construct efficient Wegman-Carter type authentication codes based on certain types of universal hash families.

Universal Hashing

For a hash function h, and for $x_1, x_2 \in \mathcal{X}$, $x_1 \neq x_2$, define $\delta_h(x_1, x_2) = 1$ if $h(x_1) = h(x_2)$, and $\delta_h(x_1, x_2) = 0$ otherwise. For a finite set $\mathcal{H}$ of hash functions, define

$$\delta_{\mathcal{H}}(x_1, x_2) = \sum_{h \in \mathcal{H}} \delta_h(x_1, x_2).$$

Let ϵ be a positive real number. $\mathcal{H}$ is **ϵ-almost universal** (or **ϵ-AU**) if

$$\delta_{\mathcal{H}}(x_1, x_2) \leq \epsilon |\mathcal{H}|$$

for all $x_1, x_2 \in \mathcal{X}$, $x_1 \neq x_2$.
Equivalently, $\mathcal{H}$ is ϵ-AU if $\Pr[h(x_1) = h(x_2)] \leq \epsilon$, where the probability is computed over a random choice of $h \in \mathcal{H}$.

Composition Construction

Theorem

Suppose $\mathcal{H}_1$ *is an* ϵ_1*-AU class of hash functions from* $\mathcal{X}_1$ *to* $\mathcal{Y}_1$*, and suppose* $\mathcal{H}_2$ *is an* ϵ_2*-ASU class of hash functions from* $\mathcal{Y}_1$ *to* $\mathcal{Y}_2$*. Then there exists an* ϵ*-ASU class* $\mathcal{H}$ *of hash functions from* $\mathcal{X}_1$ *to* $\mathcal{Y}_2$*, where* $\epsilon = \epsilon_1 + \epsilon_2$ *and* $|\mathcal{H}| = |\mathcal{H}_1| \times |\mathcal{H}_2|$*.*

Construction. For every $h_1 \in \mathcal{H}_1$ and every $h_2 \in \mathcal{H}_2$ define a hash function $h_1 \circ h_2$ as follows:

$$(h_1 \circ h_2)(x) = h_2(h_1(x))$$

for all $x \in \mathcal{X}_1$.

Composition Construction (cont.)

Proof. Property 1. is easy to verify. We prove property 2. when $y_1 = y_2$ (this is the case that yields the highest probability).
Let $x_1, x_2 \in \mathcal{X}_1$. We distinguish two cases:

case 1 $h_1(x_1) = h_1(x_2)$ occurs with probability at most ϵ_1. Then

$$\Pr[h(x_1) = h(x_2) = y_1 | h_1(x_1) = h_1(x_2)] = \frac{1}{|\mathcal{Y}_2|}.$$

case 2 $h_1(x_1) \neq h_1(x_2)$ occurs with probability ≤ 1. Then

$$\Pr[h(x_1) = y_1 \wedge h(x_2) = y_2 | h_1(x_1) \neq h_1(x_2)] \leq \frac{\epsilon_2}{|\mathcal{Y}_2|}.$$

Combining the two cases and simplifying, we get

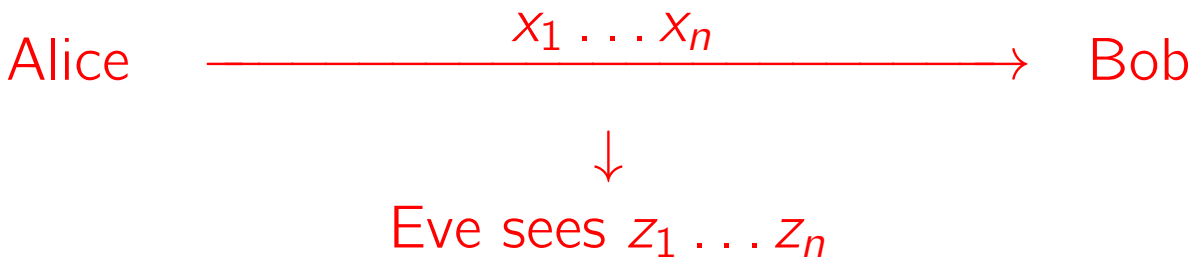
$$\Pr[h(x_1) = y_1 \wedge h(x_2) = y_2] \leq \epsilon_1 \times \frac{1}{|\mathcal{Y}_2|} + 1 \times \frac{\epsilon_2}{|\mathcal{Y}_2|} = \frac{\epsilon_1 + \epsilon_2}{|\mathcal{Y}_2|}.$$

Examples

- Recall that, in an authentication code derived from a strongly universal hash family (i.e., $\epsilon = 1/m$, the best we can hope for is
$$\log_2 |\mathcal{K}| \approx \log_2 |\mathcal{S}| + \log_2 |\mathcal{A}|.$$
- It is possible to construct $\frac{2}{m}$-ASU hash families using the composition construction that have drastically fewer keys.
- Using **Reed-Solomon codes** as ϵ-AU hash families, Bierbrauer *et al* (1993) showed that
$$\log_2 |\mathcal{K}| \approx 3 \log_2 |\mathcal{A}| + 2(\log_2 \log_2 |\mathcal{S}| - \log_2 \log_2 |\mathcal{A}|).$$

Wyner's Wiretap Channel

- The wire-tap channel was introduced in A.D. Wyner, **"The wire-tap channel"**, *The Bell System Technical Journal* **54** (1975), 1355–1387.
- Alice uses a **non-secret encoding method** to encode a one bit message b as $x = x_1 \ldots x_n$ to send to Bob.
- Suppose there is a reliable channel connecting Alice and Bob, but Eve (the eavesdropper) sees a corrupted version z of x.
- For example, suppose that Eve's wiretap channel is a **binary symmetric channel** with error probability $p < 1/2$.
- That is, $\Pr[\mathbf{z}_i = \mathbf{x}_i] = 1 - p$ and $\Pr[\mathbf{z}_i \neq \mathbf{x}_i] = p$.



Wyner's Wiretap Channel (cont.)

- To encode $b = 0$, Alice chooses a random n-tuple having even weight, and to encode $b = 1$, Alice chooses a random n-tuple having odd weight.
- Eve obtains the correct value of b if and only if there is an even number of errors, which happens with probability

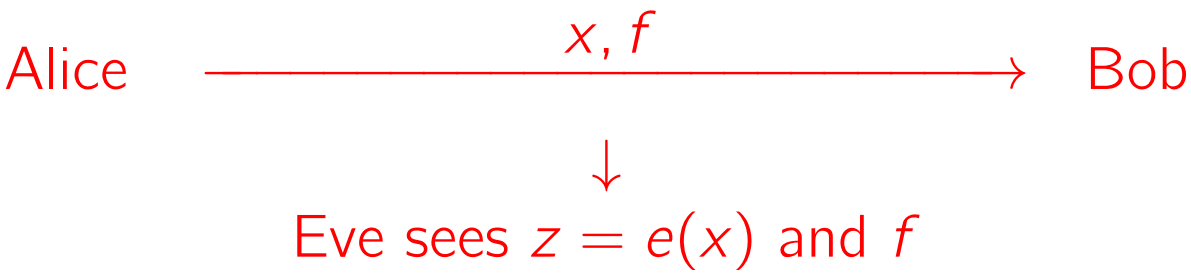
$$\frac{1}{2} + \frac{1}{2}(1 - 2p)^n.$$

- This quantity approaches $1/2$ exponentially quickly, which means that determining b from z is basically a random guess for Eve.

Privacy Amplification

- We describe a variation of the wire-tap channel from C.H. Bennett, G. Brassard, C. Crépeau and U.M. Maurer, **"Generalized privacy amplification"**, *IEEE Transactions on Information Theory* **41** (1995), 1915–1923.
- Alice transmits a message x to Bob and Eve sees $z = e(x)$, where $e : \mathcal{X} \to \mathcal{Z}$ is Eve's chosen **eavesdropping function**.
- Alice and Bob each compute $K = f(x)$, where f is a public function chosen randomly from an ϵ-AU class of hash functions from $\mathcal{X}$ to $\mathcal{Y}$, where $|\mathcal{X}| = N$ and $|\mathcal{Y}| = M$.

Privacy Amplification (cont.)



- Eve's average uncertainty about K, given f and z, is

$$I(\mathbf{K}|\mathbf{f},\mathbf{z}) = h(\mathbf{K}) - h(\mathbf{K}|\mathbf{f},\mathbf{z})$$

- This quantity is also called **mutual information** in information theory.
- It can be proven that

$$I(\mathbf{K}|\mathbf{f},\mathbf{z}) \leq \log_2 M - \log_2 N + \log_2(|\mathcal{Z}| + \epsilon N).$$

Secret Sharing

- Various types of shared control schemes depend on a cryptographic primitive called a (t, n)-**threshold scheme**.
- Let t and n be positive integers, where $t \leq n$.
- There is a trusted authority, denoted TA, and n users, denoted $U_1, \ldots, U_n$.
- The TA has a secret value $K \in \mathcal{K}$, called a **secret** or a **key**, where $\mathcal{K}$ is a specified finite set.

Secret Sharing

- The TA uses a **share generation algorithm** to split K into n **shares**, denoted $s_1, \ldots, s_n$.
- Each share $s_i \in \mathcal{S}$, where $\mathcal{S}$ is a specified finite set.
- For every i, $1 \leq i \leq n$, the share s_i is transmitted by the TA to user U_i using a secure channel.
- The following two properties should hold:
  1. a **reconstruction algorithm** can be used to reconstruct the secret, given any t of the n shares,
  2. no $t-1$ shares reveal any information as to the value of the secret.

An (n, n)-Threshold Scheme

- Suppose $K \in \mathbb{Z}_m$ is the secret.
- Let $s_1, \ldots, s_{n-1}$ be chosen independently and uniformly at random from $\mathbb{Z}_m$.
- Let

$$s_n = K - \sum_{i=1}^{n-1} s_i \bmod m.$$

- $s_1, \ldots, s_n$ are shares of an (n, n)-threshold scheme:
  1. $K = \sum s_i \bmod m$, and
  2. given all the shares except s_j, K could take on any value, depending on the value of the share s_j.

Shamir Threshold Scheme

- In 1979, Shamir showed how to construct a (t, n)-threshold scheme based on polynomial interpolation over $\mathbb{Z}_p$, where p is prime.
- Let $p \geq n + 1$ be a prime.
- Let $\mathcal{K} = \mathcal{S} = \mathbb{Z}_p$.
- In an initialization phase, $x_1, x_2, \ldots, x_n$ are defined to be n distinct non-zero elements of $\mathbb{Z}_p$.
- the TA gives x_i to U_i, for all i, $1 \leq i \leq n$.
- The x_i's are public information.

Share Generation

Protocol: **Shamir scheme share generation**

Input: A secret $K \in \mathbb{Z}_p$.

1. The TA chooses $a_1, \ldots, a_{t-1}$ independently and uniformly at random from $\mathbb{Z}_p$.
2. The TA defines

$$a(x) = K + \sum_{j=1}^{t-1} a_j x^j$$

   (note that $a(x) \in \mathbb{Z}_p[x]$ is a random polynomial of degree at most $t-1$, such that the constant term is the secret, K).
3. For $1 \leq i \leq n$, the TA constructs the share $s_i = a(x_i)$ and gives it to U_i using a secure channel.

Reconstruction

- Suppose users $U_{i_1}, \ldots, U_{i_t}$ want to determine K.
- They know that $s_{i_j} = a(x_{i_j})$, $1 \le j \le t$.
- Since $a(x)$ is a polynomial of degree at most $t-1$, they can determine $a(x)$ by Lagrange interpolation; then $K = a(0)$.
- The **Lagrange interpolation formula** is as follows:

$$a(x) = \sum_{j=1}^{t} s_{i_j} \prod_{1 \le k \le t, k \ne j} \frac{x - x_{i_k}}{x_{i_j} - x_{i_k}}.$$

- set $x = 0$; then

$$K = \sum_{j=1}^{t} s_{i_j} \prod_{1 \le k \le t, k \ne j} \frac{-x_{i_k}}{x_{i_j} - x_{i_k}} = \sum_{j=1}^{t} s_{i_j} \prod_{1 \le k \le t, k \ne j} \frac{x_{i_k}}{x_{i_k} - x_{i_j}}.$$

Reconstruction (cont.)

Protocol: **Shamir scheme secret reconstruction**

Input: $x_{i_1}, \ldots, x_{i_t}$, $s_{i_1}, \ldots, s_{i_t}$

1. For $1 \leq j \leq t$, define

$$b_j = \prod_{1 \leq k \leq t, k \neq j} \frac{x_{i_k}}{x_{i_k} - x_{i_j}}.$$

   Note: the b_j's do not depend on the shares, so they can be precomputed (for a given subset of t users).

2. Compute

$$K = \sum_{j=1}^{t} b_j \, s_{i_j}.$$

Example

- Suppose that $p = 17$, $t = 3$, and $n = 5$; and the public x-co-ordinates are $x_i = i$, $1 \leq i \leq 5$.
- Suppose that the participants in $G = \{U_1, U_3, U_5\}$ wish to compute K, given their shares 8, 10 and 11, respectively.
- The following computations are performed:

$$
\begin{aligned}
b_1 &= \frac{x_3 x_5}{(x_3 - x_1)(x_5 - x_1)} \bmod 17 \\
&= 3 \times 5 \times (-2)^{-1} \times (-4)^{-1} \bmod 17 \\
&= 4, \\
b_2 &= 3, \quad \text{and} \\
b_3 &= 11 \\
K &= 4 \times 8 + 3 \times 10 + 11 \times 11 \bmod 17 = 13.
\end{aligned}
$$

Security of the Shamir Scheme

- Suppose users $U_{i_1}, \ldots, U_{i_{t-1}}$ want to determine K.
- They know that $s_{i_j} = a(x_{i_j})$, $1 \leq j \leq t-1$.
- Let K_0 be arbitrary.
- By **Lagrange interpolation**, there is a unique polynomial $a_0(x)$ such that $s_{i_j} = a_0(x_{i_j})$ for $1 \leq j \leq t-1$ and such that $K_0 = a_0(0)$.
- Hence no value of K can be ruled out, given the shares held by $t-1$ users.

Security of the Shamir Scheme (cont.)

- With a bit more work, we can show that the **Shamir scheme** satisfies a property analogous to perfect secrecy.
- We assume an arbitrary but fixed *a priori* probability distribution on $\mathcal{K}$.
- Given any set of $\tau \leq t-1$ or fewer shares, say s_{i_j}, $j = 1, \ldots, \tau$, and given any $K_0 \in \mathcal{K}$, it is possible to show that

$$\Pr[K_0 | s_{i_1}, \ldots, s_{i_\tau}] = \Pr[K_0].$$

Key Predistribution

- **Key predistribution** refers to a protocol where a trusted authority (TA) distributes secret information to a set $\mathcal{U}$ of n users in a network.
- Each user $U \in \mathcal{U}$ receives secret information from the TA via a secure channel.
- For certain prespecified subsets $P \subseteq \mathcal{U}$, each user in P can compute a key k_P from the secret information he or she holds (no interaction is required).
- The key k_P should be secure against certain prespecified coalitions F where $F \cap P = \emptyset$.

Key Predistribution

Protocol: **Blom's key distribution scheme**

1. For $0 \leq i, j \leq k$, the *TA* chooses random elements $a_{i,j} \in \mathbb{Z}_p$, such that $a_{i,j} = a_{j,i}$ for all i, j (where k is the security parameter). Then the *TA* forms the polynomial

$$f(x, y) = \sum_{i=0}^{k} \sum_{j=0}^{k} a_{i,j}\, x^i y^j \bmod p.$$

2. For each user U, the *TA* computes the polynomial

$$g_U(x) = f(x, r_U) \bmod p = \sum_{i=0}^{k} a_{U,i}\, x^i$$

   and transmits the coefficient vector $(a_{U,0}, \ldots, a_{U,k})$ to U over a secure channel. (Every user U has a different public value $r_U \in \mathbb{Z}_p$.)

3. For any two users U and V, the key $K_{U,V} = f(r_U, r_V)$.

A Toy Example ($k = 1$)

- Suppose $p = 17$.
- Suppose there are three users: U, V and W, and their public values are $r_U = 12$, $r_V = 7$ and $r_W = 1$.
- Suppose the TA chooses the polynomial

$$f(x, y) = 8 + 7(x + y) + 2xy.$$

- the g polynomials are as follows:

$$\begin{aligned} g_U(x) &= 7 + 14x \\ g_V(x) &= 6 + 4x \\ g_W(x) &= 15 + 9x \end{aligned}$$

A Toy Example (cont.)

- the three keys are

$$\begin{aligned} K_{U,V} &= 3 \\ K_{U,W} &= 4 \\ K_{V,W} &= 10 \end{aligned}$$

- U would compute $K_{U,V}$ as

$$g_U(r_V) = 7 + 14 \times 7 \bmod 17 = 3$$

- V would compute $K_{U,V}$ as

$$g_V(r_U) = 6 + 4 \times 12 \bmod 17 = 3$$

Security of the Blom Scheme

The **Blom scheme** satisfies the following security properties:

1. No set of k users, say $W_1, \ldots, W_k$ can determine any information about a key for two other users, say $K_{U,V}$.
2. Any set of $k+1$ users, say $W_1, \ldots, W_{k+1}$, can break the scheme.

Security of the Blom Scheme (cont.)

- A set of users $W_1, \ldots, W_\ell$ (collectively) know the polynomials $g_{W_i}(x) = f(x, r_{W_i}) \bmod p$, $1 \leq i \leq \ell$.
- We use a **bivariate Lagrange interpolation formula** to prove 2.
- Suppose p is prime; $y_1, y_2, \ldots, y_{m+1} \in \mathbb{Z}_p$ are distinct; and suppose that $a_1(x), a_2(x), \ldots, a_{m+1}(x) \in \mathbb{Z}_p[x]$ are polynomials of degree at most m.
- There is a unique polynomial $A(x, y) \in \mathbb{Z}_p[x, y]$ having degree at most m (in x and y) such that $A(x, y_i) = a_i(x)$, $1 \leq i \leq m + 1$.
- The polynomial $A(x, y)$ is defined as follows:

$$A(x, y) = \sum_{j=1}^{m+1} a_j(x) \prod_{1 \leq h \leq m+1, h \neq j} \frac{y - y_h}{y_j - y_h}.$$

Example of Bivariate Interpolation

Suppose that $p = 13$, $m = 2$, $y_1 = 1$, $y_2 = 2$, $y_3 = 3$
$a_1(x) = 1 + x + x^2$, $a_2(x) = 7 + 4x^2$ and $a_3(x) = 2 + 9x$. Then:

$$\begin{aligned}
\frac{(y-2)(y-3)}{(1-2)(1-3)} &= 7y^2 + 4y + 3 \\
\frac{(y-1)(y-3)}{(2-1)(2-3)} &= 12y^2 + 4y + 10 \\
\frac{(y-1)(y-2)}{(3-1)(3-2)} &= 7y^2 + 5y + 1
\end{aligned}$$

$$\begin{aligned}
A(x,y) &= (1 + x + x^2)(7y^2 + 4y + 3) + (7 + 4x^2)(12y^2 + 4y + 10) \\
&\quad + (2 + 9x)(7y^2 + 5y + 1) \bmod 13 \\
&= y^2 + 3y + 10 + 5xy^2 + 10xy + 12x + 3x^2y^2 + 7x^2y + 4x^2
\end{aligned}$$

Insecurity wrt $k+1$ Colluders

- A set of bad users $W_1, \ldots, W_{k+1}$ (collectively) know the polynomials

$$g_{W_i}(x) = f(x, r_{W_i}) \bmod p,$$

  $1 \leq i \leq k+1$.
- Using the bivariate interpolation formula, they can compute $f(x, y)$.
- Then they can compute any key.

Security wrt k Colluders

- A set of bad users $W_1, \ldots, W_k$ (collectively) know the polynomials

$$g_{W_i}(x) = f(x, r_{W_i}) \bmod p,$$

  $1 \leq i \leq k$.
- We show that this information is consistent with any possible value of the key.
- Let K be the real (unknown) key, and let $K_0 \neq K$.
- Define a polynomial $f_0(x, y)$ as follows:

$$f_0(x, y) = f(x, y) + (K_0 - K) \prod_{1 \leq i \leq k} \frac{(x - r_{W_i})(y - r_{W_i})}{(r_U - r_{W_i})(r_V - r_{W_i})}.$$

Security wrt k Colluders (cont.)

- f_0 is a symmetric polynomial (i.e., $f_0(x, y) = f_0(y, x)$).
- For $1 \leq i \leq k$, it holds that

$$f_0(x, r_{W_i}) = f(x, r_{W_i}) = g_{W_i}(x).$$

- Further,

$$f_0(r_U, r_V) = f(r_U, r_V) + K_0 - K = K_0.$$

- We have shown that, for any possible value of the key, say K_0, there is a symmetric polynomial f_0 such that the key $K_{U,V} = K_0$ and such that the secret information held by the k bad users is unchanged.

Closing Remarks

- There is a **rich body of work** on unconditionally secure cryptography.
- Unconditionally secure schemes tend to have **simple constructions** and they are **very efficient**.
- Ultimately, any security proof in the unconditionally secure setting is a proof about certain **probability distributions**.
- Some goals **cannot** be accommodated in the unconditionally secure setting (e.g., public-key encryption schemes).
- Some goals can be achieved only by assuming **some limitations** on the usage or functionality of the scheme and/or the number of adversaries who are conspiring to break the scheme.
- Nevertheless, unconditionally secure schemes are very useful in a **wide variety of contexts**, including conventional (computationally secure) cryptography.