Better Security for Permutation-Based Tweakable Correlation-Robust Hashing

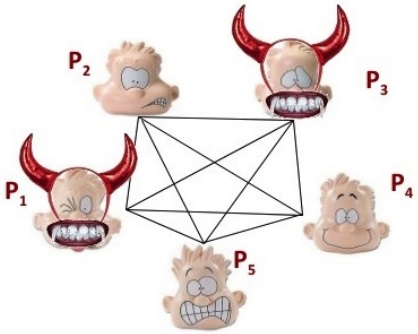
Yu Long Chen[1] Stefano Tessaro[2]

imec-COSIC, KU Leuven

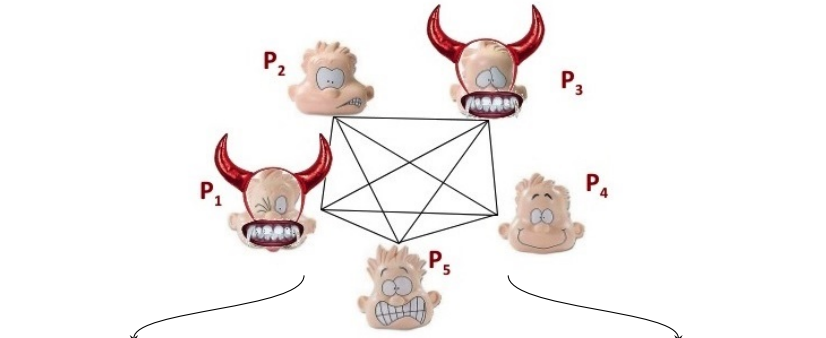
Paul G. Allen School of Computer Science & Engineering,
University of Washington

December, 2021

Symmetric-Key Primitives in Secure Multiparty Computation



Symmetric-Key Primitives in Secure Multiparty Computation



Make heavy use of symmetric-key primitives

Hash Functions as Essential Component

Special form of hash functions is needed

Hash Functions as Essential Component

Special form of hash functions is needed

- Often modeled as a Random Oracle

Hash Functions as Essential Component

Special form of hash functions is needed

- Often modeled as a Random Oracle
- Often used to hash 128-bit strings

Hash Functions as Essential Component

Special form of hash functions is needed

- Often modeled as a Random Oracle
- Often used to hash 128-bit strings
- SHA-3 has a too large state size $\rightarrow$ low performance

Hash Functions as Essential Component

Special form of hash functions is needed

- Often modeled as a Random Oracle
- Often used to hash 128-bit strings
- SHA-3 has a too large state size $\rightarrow$ low performance
- Constructions based on fixed-key AES are used to improve performance

Hash Functions as Essential Component

Special form of hash functions is needed

- Often modeled as a Random Oracle
- Often used to hash 128-bit strings
- SHA-3 has a too large state size $\rightarrow$ low performance
- Constructions based on fixed-key AES are used to improve performance

However...

Hash Functions as Essential Component

Special form of hash functions is needed

- Often modeled as a Random Oracle
- Often used to hash 128-bit strings
- SHA-3 has a too large state size $\rightarrow$ low performance
- Constructions based on fixed-key AES are used to improve performance

However...

- Many existing protocols use insecure hash functions

Hash Functions as Essential Component

Special form of hash functions is needed

- Often modeled as a Random Oracle
- Often used to hash 128-bit strings
- SHA-3 has a too large state size $\rightarrow$ low performance
- Constructions based on fixed-key AES are used to improve performance

However...

- Many existing protocols use insecure hash functions
- Several notions are needed for different protocols

Hash Functions as Essential Component

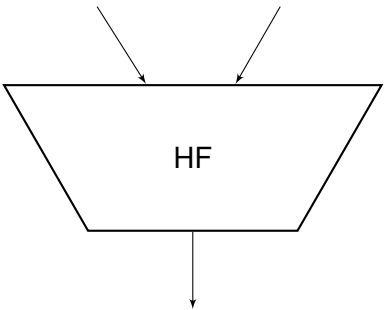
Special form of hash functions is needed

- Often modeled as a Random Oracle
- Often used to hash 128-bit strings
- SHA-3 has a too large state size $\rightarrow$ low performance
- Constructions based on fixed-key AES are used to improve performance

However...

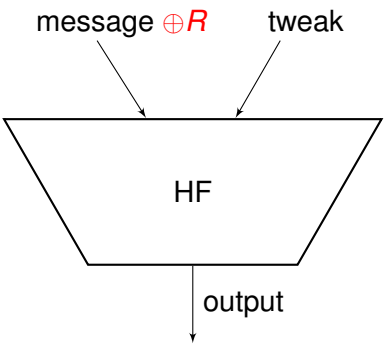
- Many existing protocols use insecure hash functions
- Several notions are needed for different protocols
- Different constructions are needed to satisfy stronger notions

Why is the Problem Hard?



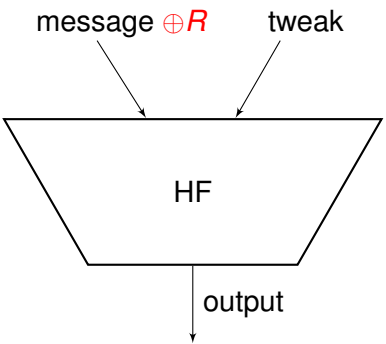
- Wanted: correlation robustness!

Why is the Problem Hard?



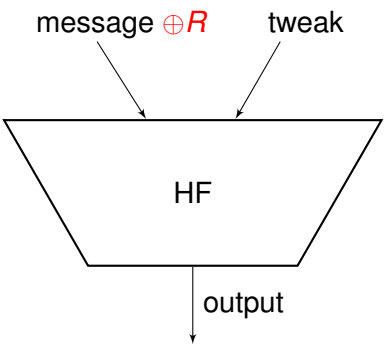
- Wanted: correlation robustness!
- Hash function $\rightarrow$ no designated secret key input

Why is the Problem Hard?



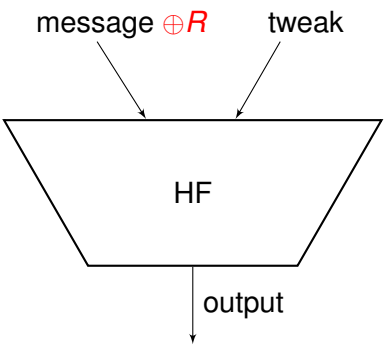
- Wanted: correlation robustness!
- Hash function $\rightarrow$ no designated secret key input
- The only randomness R is XORed to the message input

Why is the Problem Hard?



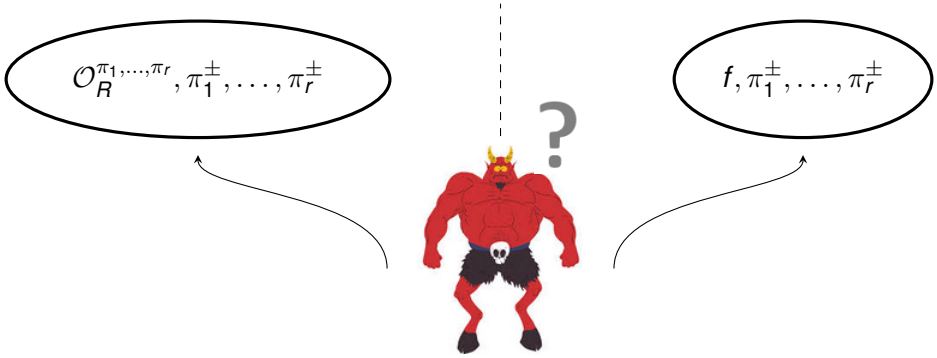
- Wanted: correlation robustness!
- Hash function $\rightarrow$ no designated secret key input
- The only randomness R is XORed to the message input
- Cannot be obtained from tweakable block ciphers such as TEM

Why is the Problem Hard?



- Wanted: correlation robustness!
- Hash function $\rightarrow$ no designated secret key input
- The only randomness R is XORed to the message input
- Cannot be obtained from tweakable block ciphers such as TEM
- Problem is related to related-key security for XOR

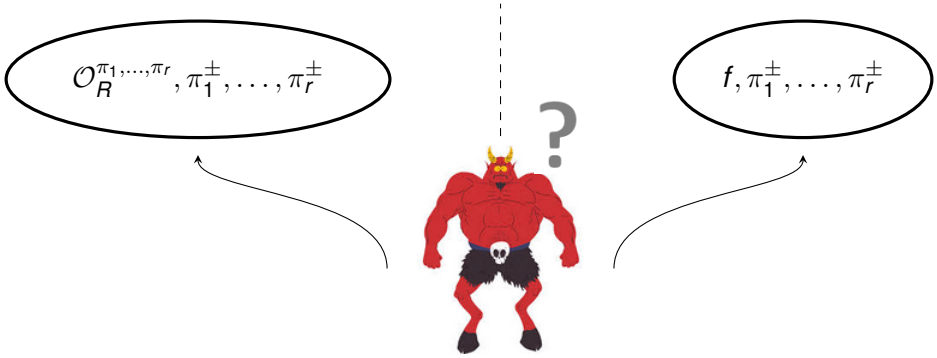
Correlation Robustness



adversary $\mathcal{A}$

- Adversary $\mathcal{A}$ makes q construction queries to oracle $\mathcal{O}_R^{\pi_1,\ldots,\pi_r}(w) = H(w \oplus R)$ or $f \xleftarrow{\$} \mathrm{Func}(n, n)$

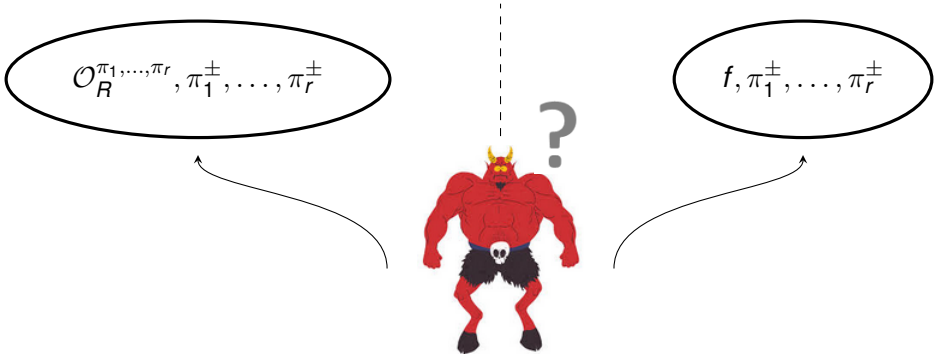
Correlation Robustness



adversary $\mathcal{A}$

- Adversary $\mathcal{A}$ makes q construction queries to oracle $\mathcal{O}_R^{\pi_1,\ldots,\pi_r}(w) = H(w \oplus R)$ or $f \xleftarrow{\$} \mathrm{Func}(n, n)$
- Adversary $\mathcal{A}$ makes p primitive queries to each of the oracle $\pi_1^{\pm}, \ldots, \pi_r^{\pm}$

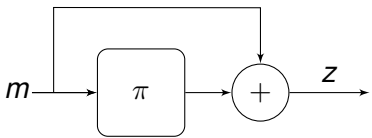
Correlation Robustness



adversary $\mathcal{A}$

- Adversary $\mathcal{A}$ makes q construction queries to oracle $\mathcal{O}_R^{\pi_1,\ldots,\pi_r}(w) = H(w \oplus R)$ or $f \xleftarrow{\$} \mathrm{Func}(n, n)$
- Adversary $\mathcal{A}$ makes p primitive queries to each of the oracle $\pi_1^{\pm}, \ldots, \pi_r^{\pm}$
- Good cr HF $\iff$ $\mathcal{A}$ cannot determine which world it is interacting with

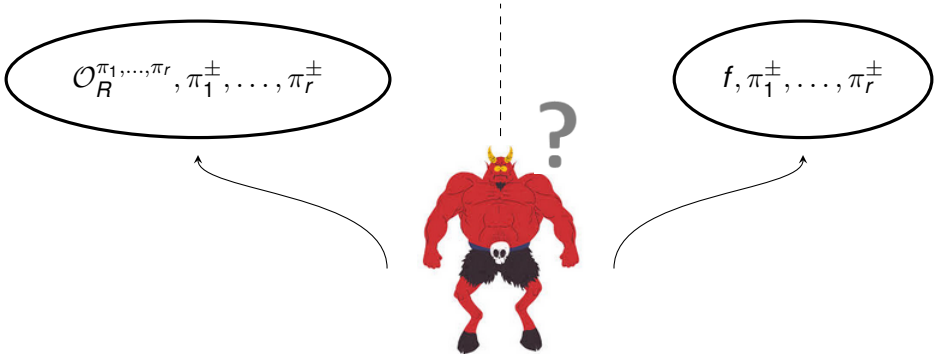
MMO Correlation Robust Hash Function



Guo, Katz, Wang, and Yu (GKWY) 2020

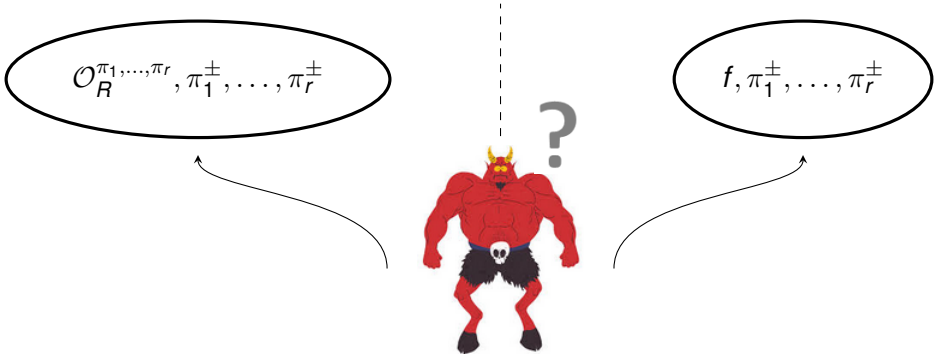
- Security bound: $O(\frac{pq}{2^n} + \frac{q^2}{2^n})$

Circular Correlation Robustness



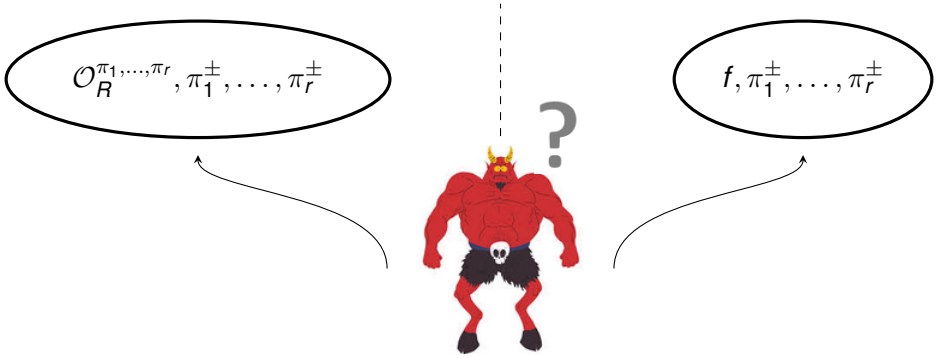
- Adversary $\mathcal{A}$ makes q construction queries to oracle $\mathcal{O}_R^{\pi_1,\ldots,\pi_r}(w, b) = H(w \oplus R) \oplus b \cdot R$ or $f \xleftarrow{\$} \mathrm{Func}(n+1, n)$

Circular Correlation Robustness



- Adversary $\mathcal{A}$ makes q construction queries to oracle $\mathcal{O}_R^{\pi_1,\ldots,\pi_r}(w, b) = H(w \oplus R) \oplus b \cdot R$ or $f \xleftarrow{\$} \mathrm{Func}(n+1, n)$
- Adversary $\mathcal{A}$ makes p primitive queries to each of the oracle $\pi_1^{\pm}, \ldots, \pi_r^{\pm}$

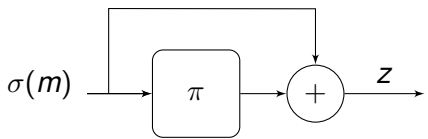
Circular Correlation Robustness



adversary $\mathcal{A}$

- Adversary $\mathcal{A}$ makes q construction queries to oracle $\mathcal{O}_R^{\pi_1,\ldots,\pi_r}(w, b) = H(w \oplus R) \oplus b \cdot R$ or $f \xleftarrow{\$} \mathrm{Func}(n+1, n)$
- Adversary $\mathcal{A}$ makes p primitive queries to each of the oracle $\pi_1^{\pm}, \ldots, \pi_r^{\pm}$
- Good ccr HF $\iff$ $\mathcal{A}$ cannot determine which world it is interacting with

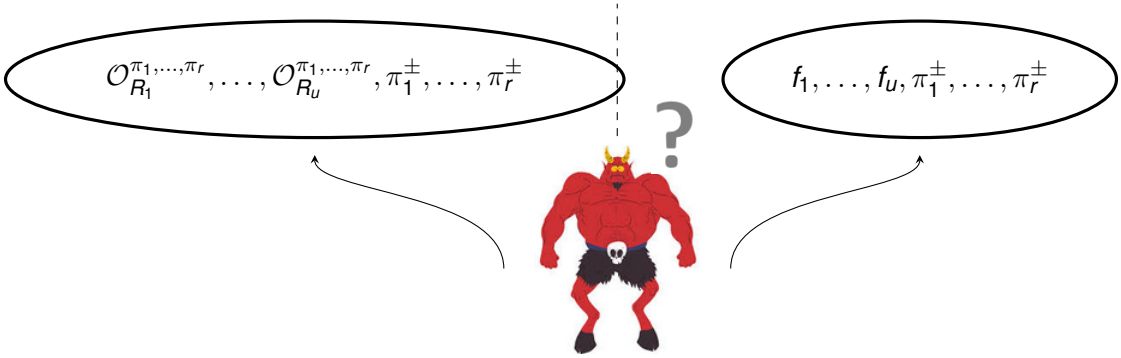
$\widehat{MMO}$ Circular Correlation Robust Hash Function



GKWY 2020

- Linear orthomorphism σ: both $\sigma(x)$ and $\sigma(x) \oplus x$ are permutations
- Security bound: $O(\frac{pq}{2^n} + \frac{q^2}{2^n})$

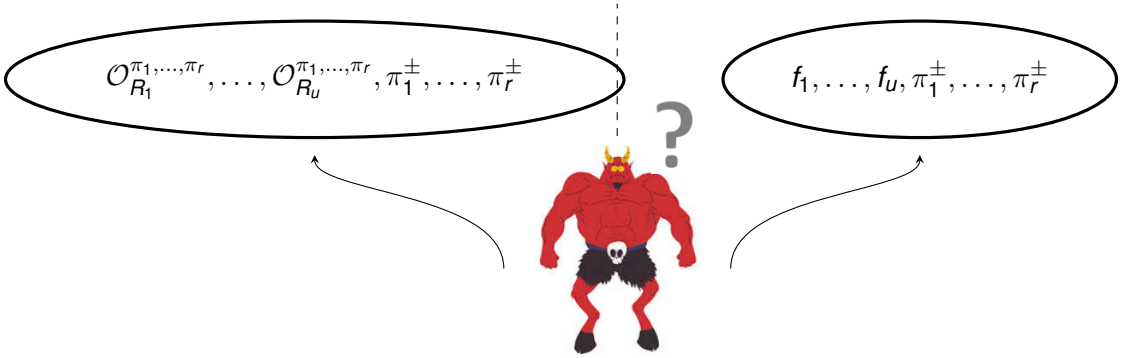
Tweakable (Circular) Correlation Robustness (Multi-Instance)



adversary $\mathcal{A}$

- Adversary $\mathcal{A}$ makes q construction queries to u oracles $\mathcal{O}_{R_1}^{\pi_1,\ldots,\pi_r}, \ldots, \mathcal{O}_{R_u}^{\pi_1,\ldots,\pi_r}$ $(\mathcal{O}_{R_i}^{\pi_1,\ldots,\pi_r}(w,t,b) = H(w \oplus R_i, t) \oplus b \cdot R_i)$ or $f_1, \ldots, f_u \xleftarrow{\$} \mathrm{Func}(n+t+1, n)$

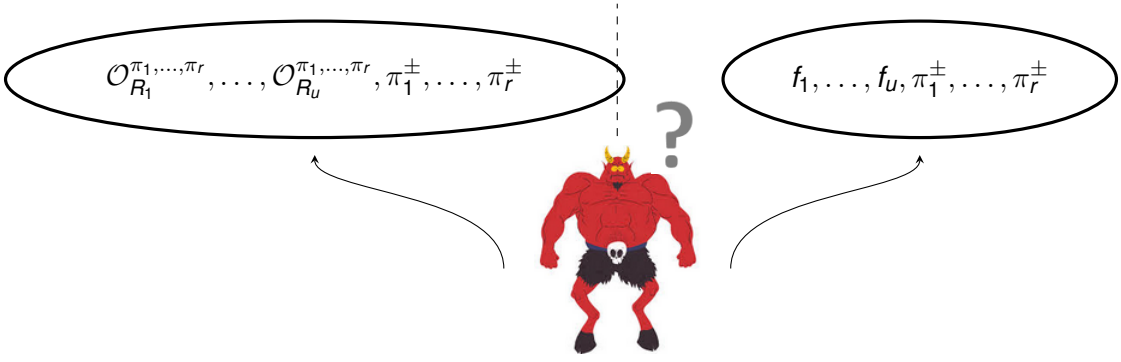
Tweakable (Circular) Correlation Robustness (Multi-Instance)



adversary $\mathcal{A}$

- Adversary $\mathcal{A}$ makes q construction queries to u oracles $\mathcal{O}_{R_1}^{\pi_1,\ldots,\pi_r}, \ldots, \mathcal{O}_{R_u}^{\pi_1,\ldots,\pi_r}$ $(\mathcal{O}_{R_i}^{\pi_1,\ldots,\pi_r}(w,t,b) = H(w \oplus R_i, t) \oplus b \cdot R_i)$ or $f_1, \ldots, f_u \xleftarrow{\$} \mathrm{Func}(n+t+1, n)$
- Adversary $\mathcal{A}$ makes p primitive queries to each of the oracle $\pi_1^{\pm}, \ldots, \pi_r^{\pm}$

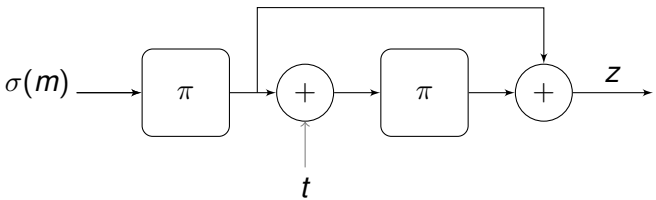
Tweakable (Circular) Correlation Robustness (Multi-Instance)



adversary $\mathcal{A}$

- Adversary $\mathcal{A}$ makes q construction queries to u oracles $\mathcal{O}_{R_1}^{\pi_1,\ldots,\pi_r}, \ldots, \mathcal{O}_{R_u}^{\pi_1,\ldots,\pi_r}$ $(\mathcal{O}_{R_i}^{\pi_1,\ldots,\pi_r}(w,t,b) = H(w \oplus R_i, t) \oplus b \cdot R_i)$ or $f_1, \ldots, f_u \xleftarrow{\$} \mathrm{Func}(n+t+1, n)$
- Adversary $\mathcal{A}$ makes p primitive queries to each of the oracle $\pi_1^{\pm}, \ldots, \pi_r^{\pm}$
- Good tccr HF $\iff$ $\mathcal{A}$ cannot determine which world it is interacting with

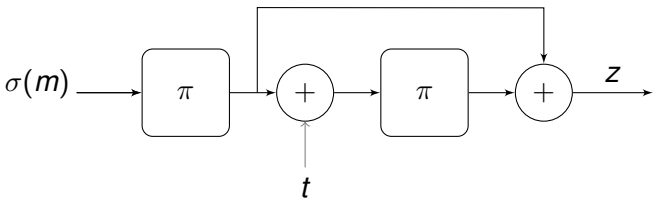
Tweakable MMO



GKWY 2020

- Security bound: $O(\frac{pq}{2^n} + \frac{q^2}{2^n})$

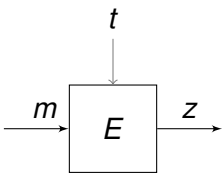
Tweakable MMO



GKWY 2020

- Security bound: $O(\frac{pq}{2^n} + \frac{q^2}{2^n})$
- Inefficient security when multi-user setting is considered (GKWWY 2020)

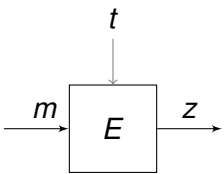
tccr Hash Function For Multi-User Setting



GKWWY 2020

- Security bound: $O(\frac{Bp}{2^n} + \frac{(B-1)q}{2^n})$

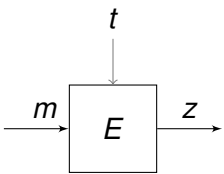
tccr Hash Function For Multi-User Setting



GKWWY 2020

- Security bound: $O(\frac{Bp}{2^n} + \frac{(B-1)q}{2^n})$
- B queries per tweak

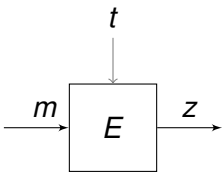
tccr Hash Function For Multi-User Setting



GKWWY 2020

- Security bound: $O(\frac{Bp}{2^n} + \frac{(B-1)q}{2^n})$
- B queries per tweak
- Multi-user security

tccr Hash Function For Multi-User Setting



GKWWY 2020

- Security bound: $O(\frac{Bp}{2^n} + \frac{(B-1)q}{2^n})$
- B queries per tweak
- Multi-user security
- Ideal cipher model

Application to OT Extension Protocols and Garbled Circuits

- Correlation robustness for semi-honest security of OT extension protocols
- Circular notion for security of free-XOR technique
- Tweakable notion for malicious security of OT extension protocols

Application to OT Extension Protocols and Garbled Circuits

- Correlation robustness for semi-honest security of OT extension protocols
- Circular notion for security of free-XOR technique
- Tweakable notion for malicious security of OT extension protocols

In this work

Application to OT Extension Protocols and Garbled Circuits

- Correlation robustness for semi-honest security of OT extension protocols
- Circular notion for security of free-XOR technique
- Tweakable notion for malicious security of OT extension protocols

In this work

- We highlight the importance for OT extension, for the first time

Application to OT Extension Protocols and Garbled Circuits

- Correlation robustness for semi-honest security of OT extension protocols
- Circular notion for security of free-XOR technique
- Tweakable notion for malicious security of OT extension protocols

In this work

- We highlight the importance for OT extension, for the first time
- Using symmetric-key building blocks
  $\rightarrow$ huge performance improvement (GKWY 2020)

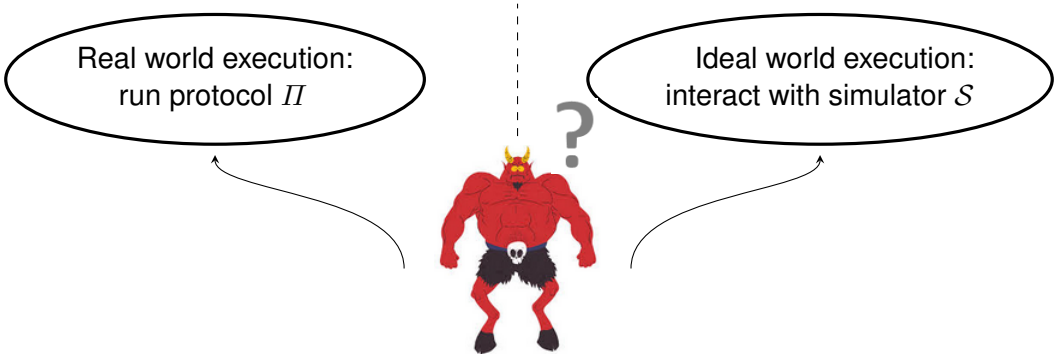
Application to OT Extension Protocols and Garbled Circuits

- Correlation robustness for semi-honest security of OT extension protocols
- Circular notion for security of free-XOR technique
- Tweakable notion for malicious security of OT extension protocols

In this work

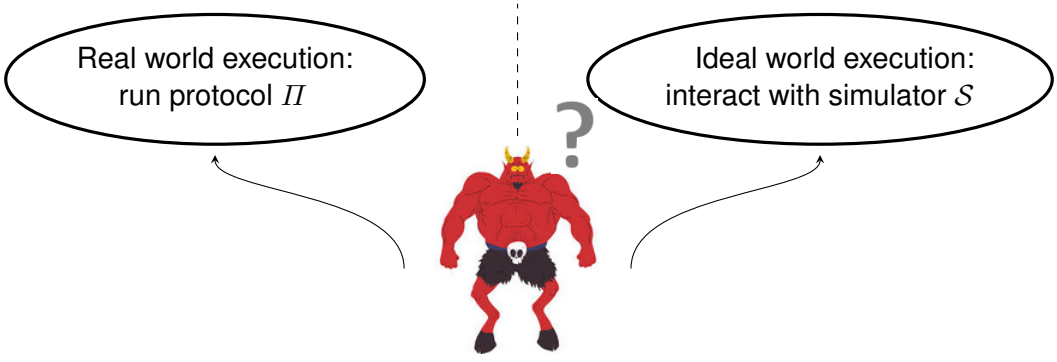
- We highlight the importance for OT extension, for the first time
- Using symmetric-key building blocks
  $\rightarrow$ huge performance improvement (GKWY 2020)
- Significant practical impact on efficiency of MPC

Security of Two-Party Computation



- Two-party hybrid-model protocol $\Pi = (\Pi_S, \Pi_R)$ accessing functionality with 3 interfaces: sender S, receiver R, and adversary $\mathcal{A}$

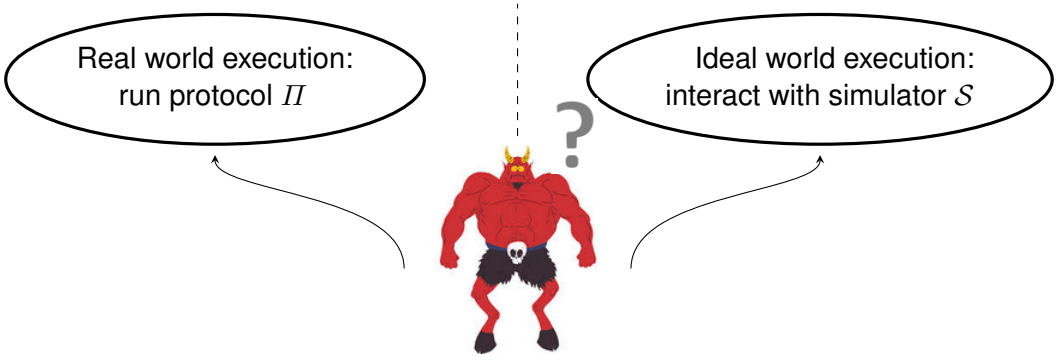
Security of Two-Party Computation



adversary $\mathcal{A}$

- Two-party hybrid-model protocol $\Pi = (\Pi_S, \Pi_R)$ accessing functionality with 3 interfaces: sender S, receiver R, and adversary $\mathcal{A}$
- At least one of the two parties is uncorrupted

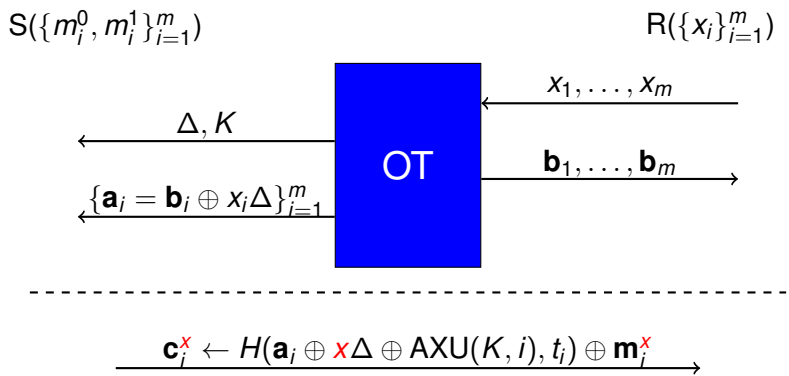
Security of Two-Party Computation



adversary $\mathcal{A}$

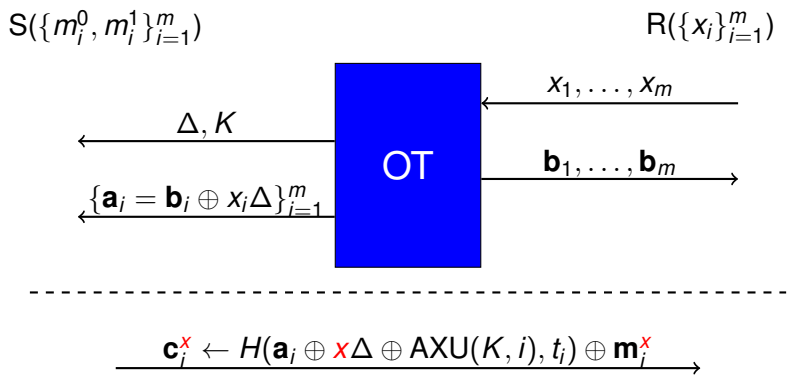
- Two-party hybrid-model protocol $\Pi = (\Pi_S, \Pi_R)$ accessing functionality with 3 interfaces: sender S, receiver R, and adversary $\mathcal{A}$
- At least one of the two parties is uncorrupted
- Secure protocol $\iff$ $\mathcal{A}$ cannot determine which world it is interacting with

Our OT Protocol



- H is only secure for distinct message inputs

Our OT Protocol



- H is only secure for distinct message inputs
- Solution: universal hash function AXU

Security of the Protocol

- Focus on sender security (corrupted receiver)

Security of the Protocol

- Focus on sender security (corrupted receiver)
- Ideal model security

Security of the Protocol

- Focus on sender security (corrupted receiver)
- Ideal model security
- Real world: $\mathbf{c}_i^{1-x_i} \leftarrow H(\mathbf{b}_i \oplus \Delta \oplus \mathrm{AXU}(K, i), t_i) \oplus \mathbf{m}_i^{x_i}$
- Ideal world: $\mathbf{c}_i^{1-x_i} \xleftarrow{\$} \{0,1\}^{\ell}$

Security of the Protocol

- Focus on sender security (corrupted receiver)
- Ideal model security
- Real world: $\mathbf{c}_i^{1-x_i} \leftarrow H(\mathbf{b}_i \oplus \Delta \oplus \mathrm{AXU}(K, i), t_i) \oplus \mathbf{m}_i^{x_i}$
- Ideal world: $\mathbf{c}_i^{1-x_i} \xleftarrow{\$} \{0,1\}^{\ell}$

Sender-security $\leq$ tcr security of H + $q^2\epsilon$

Security of the Protocol

- Focus on sender security (corrupted receiver)
- Ideal model security
- Real world: $\mathbf{c}_i^{1-x_i} \leftarrow H(\mathbf{b}_i \oplus \Delta \oplus \mathrm{AXU}(K, i), t_i) \oplus \mathbf{m}_i^{x_i}$
- Ideal world: $\mathbf{c}_i^{1-x_i} \xleftarrow{\$} \{0,1\}^{\ell}$

Sender-security $\leq$ tcr security of H + $q^2\epsilon$

- Fix a-prior a random set of tweaks T of size m

Security of the Protocol

- Focus on sender security (corrupted receiver)
- Ideal model security
- Real world: $\mathbf{c}_i^{1-x_i} \leftarrow H(\mathbf{b}_i \oplus \Delta \oplus \mathrm{AXU}(K, i), t_i) \oplus \mathbf{m}_i^{x_i}$
- Ideal world: $\mathbf{c}_i^{1-x_i} \xleftarrow{\$} \{0,1\}^\ell$

Sender-security $\leq$ tcr security of H + $q^2\epsilon$

- Fix a-prior a random set of tweaks T of size m
- AXU with $\epsilon = 1/2^n$

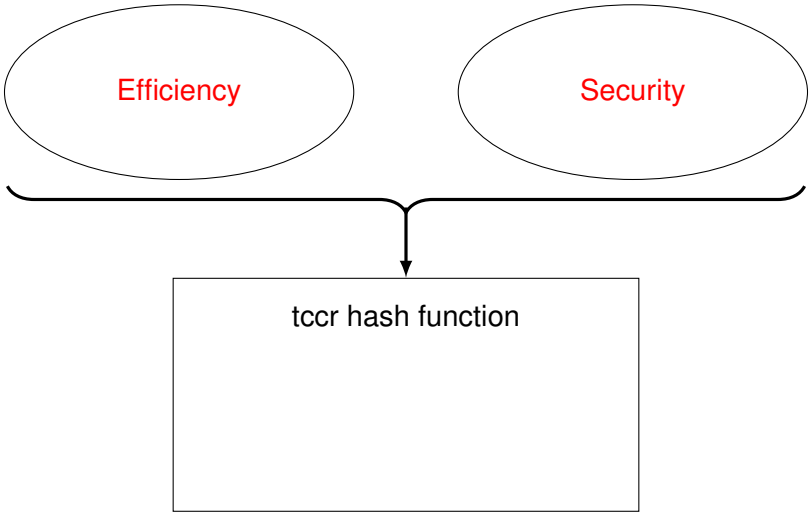
Security of the Protocol

- Focus on sender security (corrupted receiver)
- Ideal model security
- Real world: $\mathbf{c}_i^{1-x_i} \leftarrow H(\mathbf{b}_i \oplus \Delta \oplus \mathrm{AXU}(K, i), t_i) \oplus \mathbf{m}_i^{x_i}$
- Ideal world: $\mathbf{c}_i^{1-x_i} \xleftarrow{\$} \{0,1\}^{\ell}$

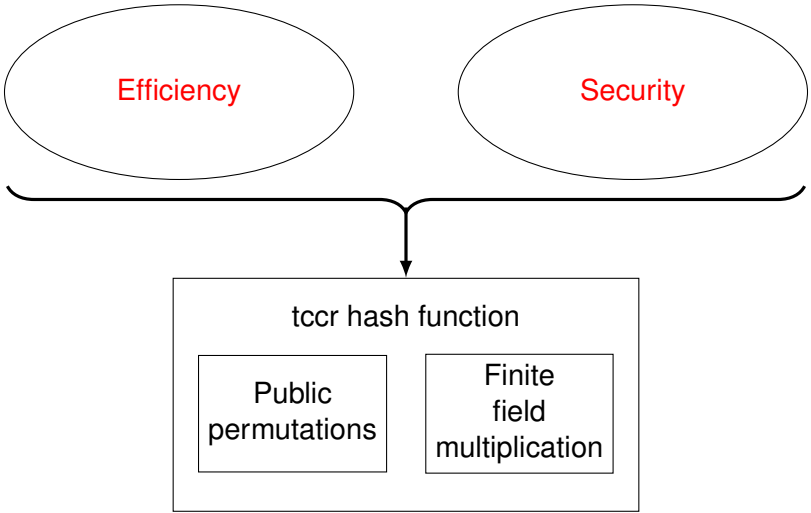
Sender-security $\leq$ tcr security of H + $q^2\epsilon$

- Fix a-prior a random set of tweaks T of size m
- AXU with $\epsilon = 1/2^n$
- Security dominated by $O(\sqrt{m}p/2^n)$

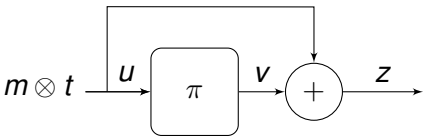
Our Constructions



Our Constructions

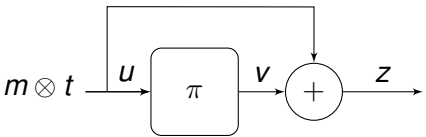


tccr Hash Function Based on One Permutation Call



Security bound: $O(\frac{pq}{2^n} + \frac{q^2}{2^n})$

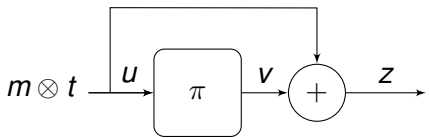
tccr Hash Function Based on One Permutation Call



Security bound: $O(\frac{pq}{2^n} + \frac{q^2}{2^n})$

- Construction query (m, t, z) and primitive query $(u, v) \Rightarrow m \otimes t = u$

tccr Hash Function Based on One Permutation Call

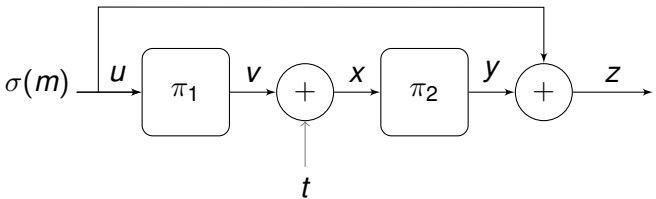


Security bound: $O(\frac{pq}{2^n} + \frac{q^2}{2^n})$

- Construction query (m, t, z) and primitive query $(u, v) \Rightarrow m \otimes t = u$
- Construction query inputs (m, t, z) and $(m', t', z') \Rightarrow m \otimes t = m' \otimes t'$

tccr Hash Function Based on Two Permutation Calls

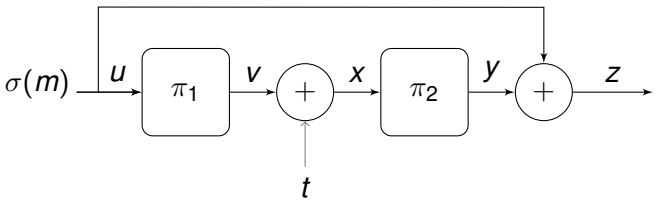
FPTP (Feed-forward Permutation-Tweak-Permutation)



Security bound: $O(\frac{B\sqrt{qp}}{2^n} + \frac{q^2}{2^n})$

tccr Hash Function Based on Two Permutation Calls

FPTP (Feed-forward Permutation-Tweak-Permutation)

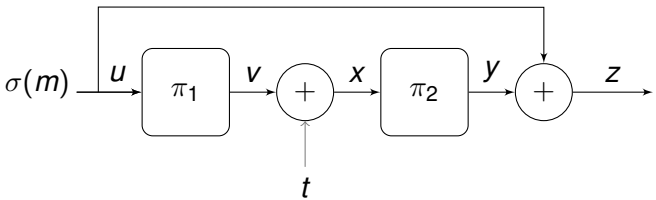


Security bound: $O(\frac{B\sqrt{qp}}{2^n} + \frac{q^2}{2^n})$

- B queries per tweak

tccr Hash Function Based on Two Permutation Calls

FPTP (Feed-forward Permutation-Tweak-Permutation)

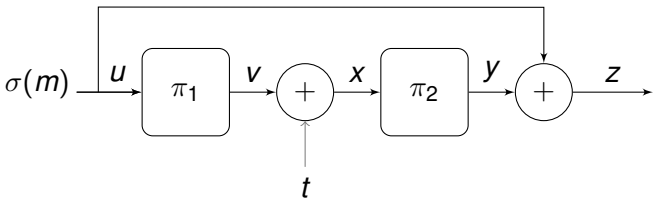


Security bound: $O(\frac{B\sqrt{qp}}{2^n} + \frac{q^2}{2^n})$

- B queries per tweak
- Tweaks chosen from nice combinatorial subset $T \subseteq \{0,1\}^n$ (e.g. random subset)

tccr Hash Function Based on Two Permutation Calls

FPTP (Feed-forward Permutation-Tweak-Permutation)

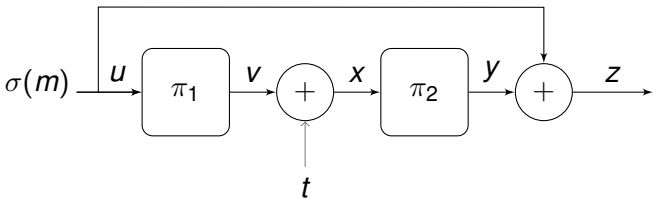


Security bound: $O(\frac{B\sqrt{qp}}{2^n} + \frac{q^2}{2^n})$

- B queries per tweak
- Tweaks chosen from nice combinatorial subset $T \subseteq \{0,1\}^n$ (e.g. random subset)
- Distinct and uniform input messages

tccr Hash Function Based on Two Permutation Calls

FPTP (Feed-forward Permutation-Tweak-Permutation)

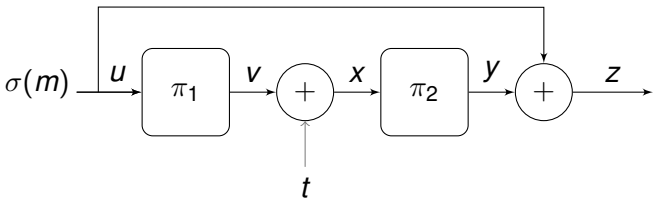


Security bound: $O(\frac{B\sqrt{qp}}{2^n} + \frac{q^2}{2^n})$

- B queries per tweak
- Tweaks chosen from nice combinatorial subset $T \subseteq \{0,1\}^n$ (e.g. random subset)
- Distinct and uniform input messages
- Arbitrary input messages using $m \otimes t$ instead of m

tccr Hash Function Based on Two Permutation Calls

FPTP (Feed-forward Permutation-Tweak-Permutation)

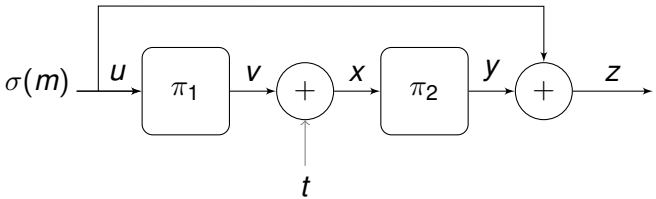


Security bound: $O(\frac{B\sqrt{qp}}{2^n} + \frac{q^2}{2^n})$

- B queries per tweak
- Tweaks chosen from nice combinatorial subset $T \subseteq \{0,1\}^n$ (e.g. random subset)
- Distinct and uniform input messages
- Arbitrary input messages using $m \otimes t$ instead of m
- Both for two independent and same permutation calls

tccr Hash Function Based on Two Permutation Calls

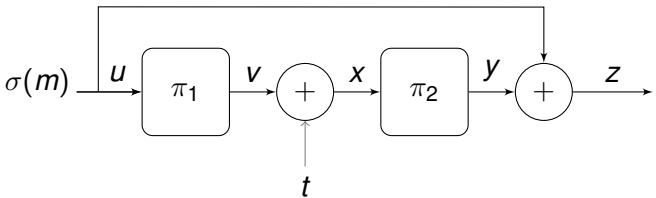
FPTP (Feed-forward Permutation-Tweak-Permutation)



Security bound: $O(\frac{B\sqrt{qp}}{2^n} + \frac{q^2}{2^n})$

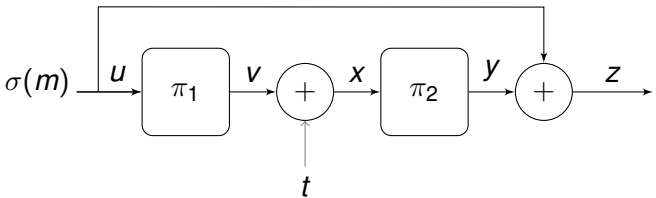
- B queries per tweak
- Tweaks chosen from nice combinatorial subset $T \subseteq \{0,1\}^n$ (e.g. random subset)
- Distinct and uniform input messages
- Arbitrary input messages using $m \otimes t$ instead of m
- Both for two independent and same permutation calls
- Multi-user security

FPTP (technical overview)



Security bound: $O(\frac{B\sqrt{\bar{q}p}}{2^n} + \frac{q^2}{2^n})$

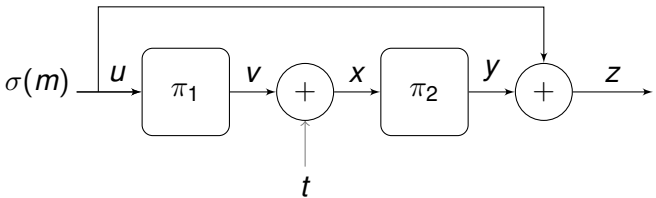
FPTP (technical overview)



Security bound: $O(\frac{B\sqrt{qp}}{2^n} + \frac{q^2}{2^n})$

- Chains: construction query (m, t, z), and primitive queries (u, v) and (x, y)
  $\Rightarrow \sigma(m) = u$ and $v \oplus t = x$

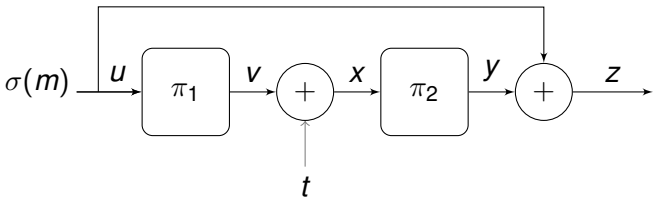
FPTP (technical overview)



Security bound: $O(\frac{B\sqrt{qp}}{2^n} + \frac{q^2}{2^n})$

- Chains: construction query (m, t, z), and primitive queries (u, v) and (x, y) $\Rightarrow \sigma(m) = u$ and $v \oplus t = x$
- Double-chains: construction query (m, t, z), and primitive queries (u, v) and (x, y) $\Rightarrow m = u$ and $y = \sigma(m) \oplus z$

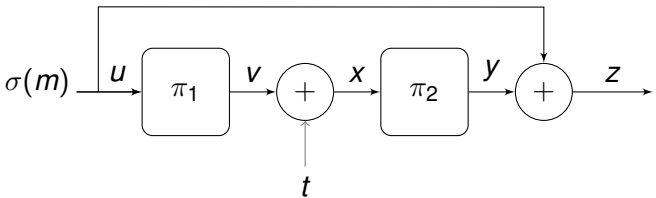
FPTP (technical overview)



Security bound: $O(\frac{B\sqrt{\bar{q}p}}{2^n} + \frac{q^2}{2^n})$

- Chains: construction query (m, t, z), and primitive queries (u, v) and (x, y) $\Rightarrow \sigma(m) = u$ and $v \oplus t = x$
- Double-chains: construction query (m, t, z), and primitive queries (u, v) and (x, y) $\Rightarrow m = u$ and $y = \sigma(m) \oplus z$
- Merging chains: construction queries (m, t, z) and (m', t', z'), and primitive queries (u, v) and $(u', v') \Rightarrow \sigma(m) = u$ and $\sigma(m') = u'$ and $v \oplus t = v' \oplus t'$

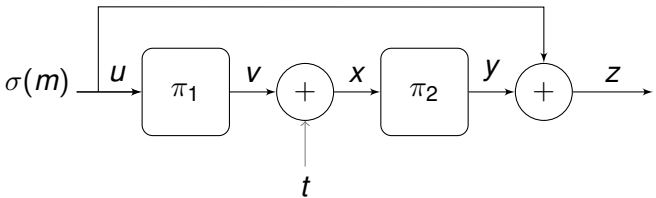
FPTP (technical overview)



Security bound: $O(\frac{B\sqrt{\bar{q}p}}{2^n} + \frac{q^2}{2^n})$

- Chains: construction query (m, t, z), and primitive queries (u, v) and (x, y) $\Rightarrow \sigma(m) = u$ and $v \oplus t = x$
- Double-chains: construction query (m, t, z), and primitive queries (u, v) and (x, y) $\Rightarrow m = u$ and $y = \sigma(m) \oplus z$
- Merging chains: construction queries (m, t, z) and (m', t', z'), and primitive queries (u, v) and $(u', v') \Rightarrow \sigma(m) = u$ and $\sigma(m') = u'$ and $v \oplus t = v' \oplus t'$
- Sum-capture theorems (Babai 02, Steinberger 12, Cogliati and Seurin 18)

FPTP (technical overview)



Security bound: $O(\frac{B\sqrt{\bar{q}p}}{2^n} + \frac{q^2}{2^n})$

- Chains: construction query (m, t, z), and primitive queries (u, v) and (x, y) $\Rightarrow \sigma(m) = u$ and $v \oplus t = x$
- Double-chains: construction query (m, t, z), and primitive queries (u, v) and (x, y) $\Rightarrow m = u$ and $y = \sigma(m) \oplus z$
- Merging chains: construction queries (m, t, z) and (m', t', z'), and primitive queries (u, v) and $(u', v') \Rightarrow \sigma(m) = u$ and $\sigma(m') = u'$ and $v \oplus t = v' \oplus t'$
- Sum-capture theorems (Babai 02, Steinberger 12, Cogliati and Seurin 18)
- New balls-into-bins combinatorial lemmas

Conclusion

New results

- Concrete security treatment of oblivious transfer extension
- $O(qp/2^n + q^2/2^n)$ secure tccr HF using one permutation call
- $O(\sqrt{q}p/2^n + q^2/2^n)$ secure tccr HF (FPTP) using two permutation calls

Conclusion

New results

- Concrete security treatment of oblivious transfer extension
- $O(qp/2^n + q^2/2^n)$ secure tccr HF using one permutation call
- $O(\sqrt{q}p/2^n + q^2/2^n)$ secure tccr HF (FPTP) using two permutation calls

Future research

- Improve the security of FPTP by improving $q^2/2^n$ to $q^3/2^{2n}$
- Formalize the limitation of the tweak set used in FPTP
- Extend FPTP to multiple rounds

Conclusion

New results

- Concrete security treatment of oblivious transfer extension
- $O(qp/2^n + q^2/2^n)$ secure tccr HF using one permutation call
- $O(\sqrt{q}p/2^n + q^2/2^n)$ secure tccr HF (FPTP) using two permutation calls

Future research

- Improve the security of FPTP by improving $q^2/2^n$ to $q^3/2^{2n}$
- Formalize the limitation of the tweak set used in FPTP
- Extend FPTP to multiple rounds

Thank you for your attention!